**Nos. 2014-1629, -1630**

# United States Court of Appeals for the Federal Circuit

NUANCE COMMUNICATIONS, INC., a Delaware Corporation,

*Plaintiff-Appellant*,

v.

ABBYY USA SOFTWARE HOUSE, INC., a California Corporation,
ABBYY SOFTWARE, LTD., a Cyprus Corporation,
ABBYY PRODUCTION LLC, a Russia Corporation, and
LEXMARK INTERNATIONAL, INC., a Delaware Corporation,

*Defendants-Cross-Appellants.*

Appeals from the United States District Court for the Northern District of California, case no. 3:08-cv-02912-JSW, Judge Jeffrey S. White

**BRIEF FOR PLAINTIFF-APPELLANT NUANCE COMMUNICATIONS, INC.**

MICHAEL A. JACOBS
MORRISON & FOERSTER LLP
425 Market Street
San Francisco, CA 94105
Telephone: (415) 268-7000

DEANNE E. MAYNARD
MARC A. HEARRON
JESSICA E. PALMER
MORRISON & FOERSTER LLP
2000 Pennsylvania Avenue, NW
Washington, DC 20006
Telephone: (202) 887-8740
DMaynard@mofo.com

*Counsel for Plaintiff-Appellant Nuance Communications, Inc.*

November 14, 2014

## CERTIFICATE OF INTEREST

Counsel for plaintiff-appellant Nuance Communications, Inc. certifies the following:

1. The full name of every party or amicus represented by me is:

Nuance Communications, Inc.

2. The name of the real party in interest (if the party named in the caption is not the real party in interest) represented by me is:

N/A

3. All parent corporations and any publicly held companies that own 10% or more of the stock of the party or amicus curiae represented by me are:

N/A

4. The names of all law firms and the partners or associates that appeared for the party or amicus now represented by me in the trial court or are expected to appear in this court are:

Dykema Gossett PLLC: Craig N. Hentschel

Morrison & Foerster LLP: Alison M. Tucher*, Brooks M. Beard*, Christopher J. Wiener, Danielle Coleman, Deanne E. Maynard, James P. Bennett, Jessica E. Palmer, Kelly Lowenberg, Marc A. Hearron, Michael A. Jacobs, Patrick J. Zhang*, Richard S.J. Hung

Weil, Gotshal & Manges LLP: Edward R. Reines, David J. Lender, Anish R. Desai

[Continued on next page]

Wilson Sonsini Goodrich & Rosati: Franklin M. Rubinstein, M. Craig Tyler, Ron E. Shulman*, Scott A. Sher, S. Michael Song*, Terrence J.P. Kearney*, Tung-On Kong, Woo J. Kim*, Abraham DeLaO*, Aysha A. Doman, Christopher P. Grewe, Clay Basser-Wall*, Daisy Suet Yin Poon*, Elise Marie Miller*, Henry Pan, Holly B. Baudler*, James C. Yoon, Julie M. Holloway*, Stephen R. Dartt*

* *No longer with firm*

Dated: November 14, 2014 /s/ Deanne E. Maynard

# TABLE OF CONTENTS

## TABLE OF AUTHORITIES

**Page(s)**

**CASES**

**STATUTES**



**OTHER AUTHORITIES**

## STATEMENT OF RELATED CASES

An order in this case dismissing the claims against two of the defendants-cross-appellants previously was appealed to this Court. *Nuance Commc'ns, Inc. v. ABBYY Software House*, No. 2010-1100, 626 F.3d 1222 (Fed. Cir. Nov. 12, 2010) (Rader, C.J., Newman, Prost, J.J.). Counsel for plaintiff-appellant is unaware of any other pending case that directly will affect or be affected by this Court's decision.

## JURISDICTIONAL STATEMENT

The district court had jurisdiction under 28 U.S.C. §§ 1331 and 1338. Final judgment was entered on June 23, 2014. Plaintiff-appellant Nuance Communications, Inc. ("Nuance") appealed on July 2, 2014. Defendants-cross-appellants ABBYY USA Software House, Inc., ABBYY Software, Ltd., ABBYY Production LLC, and Lexmark International, Inc. (collectively, "ABBYY") cross-appealed on July 9, 2014. This Court has jurisdiction under 28 U.S.C. § 1295(a).

## STATEMENT OF THE ISSUES

1. Whether a new trial on the asserted claims of U.S. Patent No. 6,038,342 ("'342 patent") should be granted because the district court erroneously construed "identifying" and "recognizing" in those claims by adopting a definition from a general-purpose dictionary that is contrary to the intrinsic evidence.

2. Whether after a trial that, pursuant to the district court's orders, was limited to certain claims of three patents (the '342 patent, U.S. Patent No. 5,381,489 ("'489 patent"), and U.S. Patent No. 6,742,161 ("'161 patent")), Nuance should have been permitted to pursue its claims for infringement of five other asserted patents that were never tried, abandoned, or dismissed: U.S. Patent Nos. 5,131,053 ("'053 patent"), 5,436,983 ("'983 patent"), 5,261,009 ("'009 patent"), 6,810,404 ("'404 patent"), and 6,820,094 ("'094 patent").

## INTRODUCTION

Nuance owns pioneering inventions in the field of optical character recognition ("OCR"). OCR technology is used in scanners and computers to convert, for example, a digital image of a page of text into a document that can be edited and searched on a computer. Nuance invented a number of methods and products that represented significant improvements in the quality and efficiency of OCR. The invention described in the '342 patent represented a fundamental breakthrough in OCR technology, allowing for OCR systems that can learn new characters in different fonts and therefore process documents more quickly and reliably.

Nuance brought this suit against ABBYY for infringement of eight OCR-related patents, including the '342 patent. Pursuant to the district court's narrowing orders, however, only three patents were tried. Although the jury found non-infringement on the asserted claims of those three patents, the jury was improperly instructed about the meaning of key terms in the '342 patent—"identifying" and "recognizing."

When the dispute over these claim terms arose before trial, the district court said it was "too late to do a construction" and that the court wanted "to avoid it." A1368. The court decreed that, if the parties did not "arrive at a mutually agreed upon" meaning, "then I will just either use a Black's Law Dictionary definition, or

some other definition, or just tell the jury to use its ordinary meaning." A1368-A1369. When the parties could not agree, the district court adopted a definition from a Merriam-Webster dictionary, without further explanation, construing the terms to mean "to establish the identity of." A16.

Adopting that dictionary definition divorced from the context of the patent was legal error. This Court has rejected that very approach, warning that "heavy reliance on the dictionary divorced from the intrinsic evidence risks transforming the meaning of the claim term to the artisan into the meaning of the term in the abstract, out of its particular context, which is the specification." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1321 (Fed. Cir. 2005) (en banc). In embracing a general-purpose dictionary definition, the district court disregarded that, as used in the context of the '342 patent, "identifying" or "recognizing" a character does not always mean reaching a final identification. Rather, it also includes tentative or non-final identifications. Erroneously instructed about the meaning of "identifying" and "recognizing," the jury found no infringement of the asserted claims of the '342 patent.

In addition, the district court erroneously brushed aside Nuance's right to pursue infringement of its five other patents, none of which were at issue in the trial. Throughout the litigation, the district court ordered Nuance to narrow its case. But the court repeatedly assured Nuance that it would protect Nuance's right

to pursue any patents that were not selected for the first trial. Nuance thus selected certain patent claims, reiterated that these claims and patents were not representative and would not dispose of any other claims, and alerted the court that this course could lead to piecemeal trials. Nuance never abandoned its unselected claims or agreed to their dismissal. But after the trial, the district court reversed course, ruling that the judgment entered on the jury's verdict somehow applied not only to the claims that were actually tried but to *all* of Nuance's claims. The district court did so even while acknowledging it previously had assured Nuance that Nuance's rights on its other claims would be protected.

Accordingly, a remand is warranted for a new trial on the claims of the '342 patent under the correct claim construction and for Nuance to pursue its untried claims.

## STATEMENT OF THE CASE

### A. Optical Character Recognition

OCR technology is used to discern characters in a digital image of text, such as a scanned document, and to translate the text into a format where the text may be searched or edited. A2; A47(col.1:18-25); A757. OCR systems analyze characters in the scanned image using techniques including "template matching," "feature analysis," and "context analysis." A757-A758.

Template matching involves comparing a character in the digital image with templates of known characters. A757. The OCR system determines that an unknown character matches a template if the difference between them is within a predetermined tolerance. A55(col.17:8-12); A757. Template matching can be performed relatively quickly, but its effectiveness is limited because OCR software does not contain all possible variations of characters, such as different fonts, styles, and sizes. A757. Early commercial OCR systems used only template matching and were therefore capable of reading characters only in a limited number of fonts. A2; A47(col.1:55-60); A777-A778.

The second generation of OCR systems, called "omnifont" systems, employed feature analysis, which enabled these systems to read characters in a wide variety of fonts and sizes. A2; A47(col.1:60-64); A778. Feature analysis works by examining the characteristics of an unknown character. A757. For example, a feature of the character "V" might be the angles of its lines. A537. Omnifont systems using feature analysis are more robust and reliable, and can identify many more characters, than OCR systems that perform only template matching. A757, A778. Prior art omnifont systems were much slower than their predecessors, however, because of the additional calculations that omnifont systems perform on each character. A757, A778.

A third technique used by modern OCR systems is context analysis. In some circumstances, template matching and feature analysis may narrow the possible choices for the unknown character, but ambiguities may remain. Context analysis can be used to resolve these ambiguities. A757-A758. For example, template matching and feature analysis may not determine whether a character is a lowercase "s" or an uppercase "S." A59(col.25:2-6). Context analysis involves analyzing a character's spatial context (e.g., whether it extends below the baseline and whether neighboring characters are larger or smaller) and linguistic context (e.g., whether neighboring characters are letters or numbers). A59(col.26:12-29); A757-A758.

### B. Nuance's '342 Patent

The systems and methods of the '342 patent enable the use of feature analysis in a way that matches the efficiency of systems that used only template matching. A47(col.2:1-4); A778. Nuance's invention fundamentally changed how modern OCR processing is performed by providing an OCR system that can accurately recognize characters in a wide variety of fonts while still allowing for rapid document processing. A779.

According to the '342 patent's specification, the invention first performs template matching, which attempts to match unknown characters to templates of

already-identified characters. A54(col.16:27-29). If a character is not matched to any template, the invention then performs feature analysis. A54(col.16:29-33).

The central novelty of the invention is that it uses the results of the feature-analysis process to build new templates that can be used in the template-matching process: "As one inventive aspect of the present invention, characters which are recognized by the feature analysis process are used as templates for recognition of later occurring characters." A54(col.16:35-38). The templates generated through the feature-analysis process are stored in a template "cache." A54(col.16:38-42). By using templates generated through feature analysis, the invention allows for template matching to recognize any font that is recognizable through feature analysis. A54(col.16:43-46). "By combining elements of feature analysis and template matching, the present invention offers the performance advantages of a template matching system with the omnifont characteristics of a feature analysis system." A54(col.16:47-50). As Nuance's expert Dr. Lopresti explained, the '342 patent's system is both flexible and fast, and thus "combines the best of both worlds" in OCR technology. A1827, A1918; *see* A54(col.16:46-50); A757.

At trial, Nuance asserted method claims 4, 6, 13, and 14, and apparatus claim 18 of the '342 patent. Each contains an element in which a character is "identified" or "recognized" "with" or "using" a character-recognition process. A60-A61(col.28:1-2, col.29:27-28). One of the issues in this appeal is the meaning

of "identified" and "recognized," which the parties agree have the same meaning. The issue is whether (as the district court held) identification/recognition is limited to circumstances in which the OCR system has made an unequivocal choice about a precise character or whether (as Nuance contends) it includes having narrowed the possible choices to a class of characters for further analysis.

Independent claim 4 recites:

> 4. In an optical character recognition system having a feature analysis process for identifying an unknown character, said optical character recognition system for identifying characters in a medium, a method for constructing a template library for use while processing said medium, said method comprising the steps of:
> (a) *identifying said unknown character with said feature analysis process*;
> (b) building a template for said unknown character subsequent to having identified said unknown character; and
> (c) storing said template in said template library.

A60(col.27:62-col.28:6) (emphasis added). Claim 6, which depends from claim 4, further requires "prior to said step of identifying said unknown character with said feature analysis process, performing said step of determining that said unknown character does not match a template in said template library." A60(col.28:10-14).

Independent claim 13 recites:

> 13. A method of creating a character template using an image, said image including a representation of a character, said method including the steps of:
> a) accessing said representation;

> b) analyzing said representation using a first recognition process;
> c) analyzing said scanned representation using a second recognition process, if said character is not recognized using said first recognition process;
> d) *generating a character template, if said character is recognized using said second recognition process*, and
> e) storing said character template, if a character template has been generated.

A61(col.29:17-30) (emphasis added). Claim 14 depends from claim 13, and further requires that "said first recognition process includes template matching and said second process includes feature analysis." A61(col.29:31-33).

Independent claim 18 recites a "system for recognizing a character represented on an image," said system including a processor that "appl[ies] a feature analysis process" to the unknown character "if said template matching process does not identify" the character. A61(col.30:24-35). The processor also "generat[es] a character template . . . if said feature analysis identifies" the character. A61(col.30:37-39).

Notably, the specification uses the word "recognized" to refer not only to unequivocal identification of a character but also to the narrowing of the possible choices to "a particular class of characters." A56(col.20:20-21); *see* A56(col.20:6-9). Specifically, in a preferred embodiment, each feature-analysis routine is called an "isit" (pronounced "is it"), so named "because the routines determine whether [an unknown character] is a particular character (e.g. 'is it' an a)." A56(col.19:57-

61). If an isit recognizes a character, the output is either (1) "an ASCII code for a particular character," i.e., a computer code associated with the exact character, or (2) "a code indicating the character *is recognized as belonging to a particular class of characters*." A56(col.20:18-21) (emphasis added). If the character is not recognized at all, the output is "a reject code indicating the character is not recognized." A56(col.20:21-22); *see* A58(col.23:66-col.24:2) ("As discussed above, after completing an analysis an isit returns either the ASCII code or the shape code for the particular image or information identifying the reason why the image was rejected.").

The specification goes on to explain the difference in meaning between the codes that may be output once a character is recognized. "An output of the ASCII code for a particular character indicates the character's identification by the isit routine is unequivocal." A56(col.20:22-24). But an isit may "recognize" a character without making an unequivocal identification because multiple "characters sharing the same topography are *recognized* by a single isit." A56(col.20:6-7) (emphasis added). "For example, the letter lower case 'p' and the letter upper case 'P' are *recognized* by the same isit." A56(col.20:7-9) (emphasis added). Similarly, isits may recognize a character as an "s" or "S," an "O" (capital O) or "0" (zero), or a comma or apostrophe. A56(col.20:9-13), A59(col.25:6-7). When such an equivocal identification is made, the output is not the ASCII code

but rather "a code indicating the character is *recognized* as belonging to a particular class of characters." A56(col.20:19-21) (emphasis added). Such a code narrows the further analysis to be performed on the character. A56(col.20:25-28), A59(col.25:7-10). This additional analysis may include further feature analysis and context analysis. A56(col.20:25-28), A59(col.25:1-11).

## C. Claim Construction

### ***1. The parties initially agreed that "identifying" to a skilled artisan does not require an unambiguous identification***

The initial dispute over the meaning of "identifying" and "recognizing" concerned whether, as ABBYY contended, "identifying" could be performed only through specific methods of character recognition. Specifically, ABBYY sought to limit "identifying" to "[d]etermining the identity of an unknown character through feature analysis or template matching or both." A9. Nuance contended that ABBYY's narrow construction contradicted the commonly understood meaning of "identifying" as used in the patent. A8-A9; A519-A521. Nuance thus asserted that "identifying" should have its plain and ordinary meaning to a person of skill in the art.

The district court rejected ABBYY's attempt to limit "identifying," reasoning that "identifying" "appears throughout portions of the patents that do not allude to the specific processing and may be analyzed according to any one of a number of techniques." A9. The court did "not find it persuasive to limit the

construction of the term to only either feature analysis [or] template analysis." A9. Accordingly, the district court "construe[d] the term 'identifying' to mean: 'identifying.'" A9.

At that time, the parties agreed that the ordinary meaning of "identifying" to a person skilled in the art does not require an unambiguous identification. That is confirmed by ABBYY's initial claim-construction brief. In that brief, ABBYY explained that the "product of [feature] analysis is *identification* of the unknown character." A537 (emphasis added). ABBYY went on to explain that "[m]any times *an identified character is still ambiguous*. For example, is it an 's' or an 'S'? Each has similar features. Contextual analysis resolves these ambiguities through a number of different ways." A537 (emphasis added).

#### ***2. When a dispute arose before trial, the district court stated "it's too late to do a construction. . . . And if the parties don't agree [to a construction], then I will just either use a Black's Law Dictionary definition, or some other definition"***

Before trial, ABBYY moved for summary judgment of non-infringement, including as to the asserted claims of the '342 patent. In the course of summary-judgment briefing, it became clear that there was a new, unresolved disagreement between the parties over the meaning of "identifying" and "recognizing."

Without pointing to any intrinsic evidence to support its new construction, ABBYY argued that "identifying" and "recognizing" require the unequivocal identification of a specific character. A941-A944, A966-A967. According to

ABBYY, the use of feature analysis in an OCR system does not result in identification or recognition within the meaning of the patent if the feature-analysis process simply identifies possible choices for the unknown character and relies on context analysis to make the final decision. A943-A944, A966-A967.

Nuance responded that ABBYY's new, narrow reading was contrary to the specification, which teaches that "identifying" and "recognizing" can encompass ambiguous identifications. A996-A998. Nuance also pointed out that ABBYY's expert confirmed that "identifying" an unknown character does not require "unambiguously identifying" the character because ambiguities can exist after identifying the unknown character, which can be resolved through context analysis. A999-A1000.

The district court denied ABBYY's motion for summary judgment of non-infringement, stating only that "there remain questions of fact regarding each of the patent infringement claims." A1122.

Shortly thereafter, Nuance requested the opportunity to brief the meaning of "identify" and "recognize" so that the parties' dispute could be resolved before trial. A1127-A1130. Nuance contended that "the existence of outstanding disputes regarding claim construction only recently arose as part of summary judgment briefing, and disputes regarding claim construction and the proper scope

of claims should be resolved by the Court, not a jury." A1127. ABBYY opposed, arguing that "identifying" already had been construed. A1134-A1135.

The district court ordered the parties "to arrive at a mutually agreed upon -- quote, unquote -- 'ordinary meaning'" of "identifying" and "recognizing." A1368. The district court stated: "I think it's too late to do a construction. I think it's unnecessary. I'd like to avoid it. And if the parties don't agree, then I will just either use a Black's Law Dictionary definition, or some other definition, or just tell the jury to use its ordinary meaning." A1368-A1369.

#### *3. When the parties could not agree, the district court disregarded the specification and adopted a contrary definition from Merriam-Webster's Ninth New Collegiate Dictionary*

Nuance and ABBYY could not agree on a construction. Nuance proposed that "identifying" be construed as "[i]dentifying (finally or tentatively)." A1436. Alternatively, Nuance proposed the following construction: "'Identifying' has its plain and ordinary meaning. Many times an identified character is still ambiguous." A1436. The second sentence of Nuance's alternative construction was taken directly from ABBYY's own earlier claim-construction briefing, in which ABBYY expressly had acknowledged that "[m]any times an identified character is still ambiguous." A537. But pre-trial, ABBYY instead offered definitions of "identify" and "recognize" from general-purpose dictionaries. A1438.

Without explanation, the district court adopted one of the definitions of "identify" from Merriam-Webster's Ninth New Collegiate Dictionary: "to establish the identity of." A16.

### D. Trial Proceedings

#### *1. The district court instructed the parties "to narrow down the claims that are initially adjudicated," while acknowledging that Nuance has "a constitutional right to have their claim[s] adjudicated"*

Throughout the proceedings, the district court required Nuance to select a subset of its patent claims to be included in the first trial. All along the way, the district court repeatedly assured Nuance that it would have the opportunity later to litigate its unselected claims.

Specifically, at a case-management conference early in the litigation, the district court instructed the parties to meet and confer "on a way to narrow down the claims that are *initially* adjudicated." A384 (emphasis added). The district court explained that the "overall driver" of the need to reduce the number of claims is that the court would not construe more than 10 claim terms in a single phase of claim construction. A373-A374, A385. ABBYY acknowledged at that conference that any narrowing of the claims must be "without prejudice to any additional claims being litigated thereafter." A375.

After meeting and conferring, Nuance proposed that it would proceed on five selected patents related to OCR technology "at this time" and "postpone

resolution of its infringement case" as to the three non-OCR patents. A401-A403. ABBYY proposed that the court sever the case, with the five OCR-related patents in one case and the three other patents in the second case. A405.

At the next case-management conference, Nuance emphasized: "We are not suggesting that we believe we're willing to withdraw these patents from the case . . . ." A419. Rather, Nuance was agreeing simply to put certain patents "on the side burner for now" so that those patents "are not going to be in the *first* trial." A418-A420 (emphasis added). Nuance explicitly cautioned that its approach may result in "two trials." A421.

The district court acknowledged in the hearing that Nuance has "a constitutional right to have their claim[s] adjudicated." A435. The court further explained: "All we're doing is saying, okay, we'll have a smaller case now or set of cases then down the road. We'll have to face the next case because I can't deny them their rights, Nuance people's rights to ultimately litigate." A449.

At the next case-management conference, after considering the recommendation of a special master, the district court ordered that "the next phase of this case" be limited to nine claims of the '342, '489, and '009 patents. A464. This limitation would apply "at least[] as far as the *Markman* procedure, *Markman* proceedings is concerned." A464. The district court assured Nuance that it would have the opportunity to proceed on its remaining claims: "I understand what my

obligations are. Nobody is going to be denied due process. I remember you said you had [a] constitutional right to have your patents heard . . . . I'm going to give you an opportunity, I'm not going to say arbitrarily, I could state, but probably get reversed for it, we're only going to do this patent." A477. The district court never gave Nuance any indication that selecting certain claims for trial would constitute abandoning the unselected claims.

After the initial claim construction, ABBYY proposed in a joint case-management report that the case "continue to focus on" the '342, '489, and '009 patents, "as these patents are all case dispositive and should be dealt with first to conserve judicial resources." A660-A661. Nuance disputed that these three patents were case-dispositive and reiterated in that same joint report that Nuance "has no intention or desire to drop its claims on the other five patents that it believes are being infringed by ABBYY and Lexmark based on the resolution of the three patents picked . . . to be construed by the Court first." A659-A660, A665-A666.

In the interests of making the case more manageable, and as an alternative to ABBYY's proposal to proceed only on the three above patents, Nuance proposed that after discovery it "select 4 patents and 15 claims for proceeding with expert discovery and trial." A665. Nuance noted the possibility that this course could lead to a second trial: "If the case is unable to be resolved before trial, it is only

fair that Nuance—not any of the Defendants—be able to select the patents for the *first (and hopefully only) trial in this case*." A666 (emphasis added). The district court adopted this proposal without indicating that adjudication of the selected claims would affect the unselected claims in any way. A687-A693; A723-A724.

Accordingly, Nuance selected claims from the '342, '489, '009, and '161 patents for trial, while reserving its rights on its other patents: "Nuance will narrow the case for expert discovery and trial by proceeding on the following 4 patents and 13 associated claims at this time, while reserving its rights on all other asserted patents and claims (including additional asserted but not selected claims for the selected patents)." A726. Nuance stated that it would "postpone resolution of its infringement case as to remaining asserted claims on the '342, '489, '009 and '161 patents, and all asserted claims on the" remaining patents. A726. Nuance expressly "maintain[ed] its allegations that Defendants infringe the claims in asserted but unselected claims from all previously asserted patents, and reserve[d] its right to reassert them against ABBYY and/or Lexmark at a later time in this suit or a future suit(s)." A727. In the next joint case-management report, Nuance reiterated that although it had selected a subset of claims "[p]ursuant to the Court's order," Nuance "reserves all rights on previously asserted patents and claims that were not selected as part of Nuance's court-ordered selection." A740.

Before trial, "[t]o further the Court's goal of narrowing the case for trial," Nuance "further selected a subset of eight claims from three patents" for trial: the '342, '489, and '161 patents. A1222-A1223 & n.1. Nuance again expressly "reserve[d] all rights on previously asserted patents, claims, and causes of action that were not selected as part of Nuance's selection." A1222 n.1. The district court still gave no indication that any narrowing would have any preclusive effect on Nuance.

### *2. The case proceeded to trial on only three of Nuance's patents and the district court entered judgment on the jury's non-infringement verdict*

The case therefore proceeded to trial on claims of the '342, '489, and '161 patents. With respect to the claims of the '342 patent, the infringement dispute at trial centered on whether the feature-analysis process in ABBYY's product, FineReader, identifies or recognizes an unknown character. A2095-A2102.

FineReader first uses template matching, resulting in one or more potential matches called "variants." A1895-A1896; A4151-A4152. Each variant is assigned a confidence value. A1871-A1872; A4141; A4151.

If template matching does not produce a variant with sufficiently high confidence, FineReader uses "Omnifont recognition," a type of feature analysis that analyzes character shapes. A1843-A1844, A1895-A1898; A4151-A4152. Relying on ABBYY's FineReader source code and technical documentation,

Nuance's expert Dr. Lopresti testified that the FineReader Omnifont routines may produce one or more preliminary identifications, or "guesses," with varying confidence values. A1874-A1875. For instance, they might recognize an unknown character as either a "Q" or a "0", and "would produce two possibilities, one with a higher degree of confidence, one with a lesser degree." A1874. FineReader uses context analysis to resolve these ambiguities and make a final determination. A1875-A1876, A1924-A1925; A2030-A2034; A4180, A4183.

When FineReader recognizes a new character shape, a template is created, added to the template cache, and confirmed as reliable through further testing. A4153; A4161. This process is called "cache training." A1839, A1871-A1872; A2121; A4079-A4083, A4125-A4126; A4153; A4161.

ABBYY's expert, Dr. Doermann, generally agreed with Dr. Lopresti about the operation of the FineReader software; he disagreed about whether that operation infringes. A2095-A2102. Dr. Doermann's non-infringement theory turned on the answer to the claim-construction dispute regarding whether "identifying" requires an unequivocal identification. Dr. Doermann testified that a feature analysis process yielding non-final identifications or "guesses" is not "identifying" or "recognizing" characters within the meaning of the patent's claims. A2095-A2096. According to Dr. Doermann, because the Omnifont routines may produce up to eight variants with different degrees of confidence,

they are not a feature-analysis process that identifies or recognizes characters. A2095-A2102.

After seven days of trial, the jury returned a verdict of no infringement. A3631-A3636. The district entered judgment on the jury's verdict: "Pursuant to the jury's verdict, entered on August 26, 2013, JUDGMENT is HEREBY ENTERED in favor of Defendants and against Plaintiff." A17.

#### *3. After trial, the district court ruled the judgment applied even to the untried claims, despite the court's previous and repeated assurances that Nuance would not be deprived of the opportunity to pursue all its patents*

Nuance filed post-trial motions, which the district court denied. Given that a judgment had been entered and the post-judgment motions had been resolved, Nuance filed a notice of appeal to protect its rights, and ABBYY cross-appealed. *See Nuance Commc'ns, Inc. v. ABBYY Software House*, Nos. 14-1229, -1260 (Fed. Cir.). Thereafter, the parties agreed that the district court's judgment was not final, as there were additional, unresolved claims still pending in the district court, and there had not been entry of a certification pursuant to Rule 54(b) of the Federal Rules of Civil Procedure. The parties thus jointly requested that the appeals be dismissed without prejudice to their being refiled once final judgment was entered. This Court granted that request.

Thereafter, ABBYY asked the district court to "formally dispose of all remaining claims in the case." A3742-A3743. ABBYY argued that Nuance

voluntarily abandoned all its unselected patent claims by making selections for construction and trial. A3745-A3750. Nuance opposed, explaining that it had merely postponed resolution of the unselected claims, in order to comply with the district court's case-management plan. A3847-A3849, A3851-A3852.

Notwithstanding the district court's repeated assurances throughout the litigation that Nuance would not be deprived of the opportunity to pursue claims that were not selected for trial, the district court ruled that Nuance could not pursue those claims. A22-A24. The court explained that its judgment on the verdict "in favor of Defendants and against Plaintiff" applied to the untried claims because "[t]he judgment did not exempt any of Nuance's causes of action or reserve judgment on any of Nuance's patents that it chose not to pursue at trial." A22. The district court acknowledged that "in the initial stages of this case, the Court kept the option open to Nuance to pursue discovery and claims construction on all of its originally asserted patents." A23. But the court stated (incorrectly) that "there was never any mention that there would be serial trials." A23. According to the district court, Nuance could have only one opportunity for a trial because Nuance "selected its best and strongest patents for trial after full discovery and unrestricted claims construction selection process." A23. The district court also faulted Nuance for not mentioning the existence of its remaining claims during post-trial briefing, A23-A24, even though the purpose of that briefing was to seek

judgment as a matter of law on the claims that were tried before the jury, not to address the untried claims.

## SUMMARY OF ARGUMENT

I. The district court erroneously construed "identifying" and "recognizing" by adopting a definition from a general-purpose dictionary without considering how those terms would be understood by a person of skill in the art in light of the specification. This Court has rejected that approach. While courts may consult dictionaries, the intrinsic evidence is the primary guide to the meaning of a claim term to a skilled artisan.

Here, the '342 patent's specification expressly teaches that "identifying" and "recognizing" includes making tentative identifications. According to the specification, an unknown character may be identified or recognized by narrowing the potential matches for the character to a plurality of possible matches. This meaning also is confirmed by extrinsic evidence: ABBYY's expert confirmed this meaning in a deposition, and other art in the field of OCR, including ABBYY's own patent, use "identifying" and "recognizing" as including ambiguous identifications. The judgment of non-infringement should be set aside and the case remanded for a new trial on all issues with respect to the asserted claims of the '342 patent under the correct construction.

II. Nuance asserts five patents in this litigation, and it has a right to have its infringement claims as to all of those patents adjudicated. Nevertheless, during the course of the litigation, the district court ordered Nuance to narrow its case and select for trial only a handful of its eight asserted patents. Although Nuance complied with the order and selected three patents for its first trial, Nuance never agreed that those selected patents were representative of all eight of its asserted patents. Quite the contrary, Nuance repeatedly reserved its right to pursue its unselected patents.

Despite having consistently assured Nuance that it would have the opportunity to litigate all of its claims, the district court nevertheless held that the judgment it entered on the jury's verdict of the three tried patents also applied to the unrelated patents that were never tried. The district court thus entered judgment on claims that were never adjudicated, that were not duplicative of the tried claims, and as to which it never allowed Nuance to present evidence or argument. That was an abuse of discretion. In addition, it deprived Nuance of its due process rights. Either reason is an independently sufficient basis to vacate the judgment on the untried claims and remand for Nuance to pursue those claims.

## STANDARD OF REVIEW

Claim construction is a question of law reviewed de novo. *Lighting Ballast Control LLC v. Philips Elecs. N. Am. Corp.*, 744 F.3d 1272, 1276-77 (Fed. Cir.

2014) (en banc); *Teva Pharm. USA, Inc. v. Sandoz, Inc.*, 723 F.3d 1363, 1373 (Fed. Cir. 2013), *cert. granted*, 134 S. Ct. 1761 (2014).

A "district court's litigation management decisions" will be reversed "if it abused its discretion or if the procedures deprived the litigant of due process of law." *Southern Cal. Edison Co. v. Lynch*, 307 F.3d 794, 807 (9th Cir. 2002) (citation omitted).

## ARGUMENT

### I. THE DISTRICT COURT ERRONEOUSLY CONSTRUED "IDENTIFYING" AND "RECOGNIZING" IN THE '342 PATENT

The district court should have construed "identifying" and "recognizing" as including ambiguous identifications. A new trial is required on all the asserted claims of the '342 patent under the correct construction.

#### A. The District Court Erroneously Adopted A Definition From A General-Purpose Dictionary Contrary To The Ordinary Meaning To A Skilled Artisan In Light Of The Specification

The district court adopted a definition from Merriam-Webster's Ninth New Collegiate Dictionary as the court's construction of "identify": "to establish the identity of." A16. According to the district court, this definition is "the plain and ordinary meaning of the terms 'identifying' and 'recognizing.'" A16. The district court did so without considering how the patentees used these terms in the specification. That is the opposite of how claim construction is supposed to be performed.

Indeed, this Court's decision in *Phillips* flatly rejected the district court's approach to claim construction. "Properly viewed, the 'ordinary meaning' of a claim term is its meaning to the ordinary artisan *after reading the entire patent*." 415 F.3d at 1321 (emphasis added). It is therefore a fundamental principle of claim construction that "claims 'must be read in view of the specification, of which they are a part.'" *Id.* at 1315 (quoting *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 979 (Fed. Cir. 1995) (en banc), *aff'd*, 517 U.S. 370 (1996)). The specification "'is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term.'" *Id.* (quoting *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996)).

It is axiomatic that ascertaining the ordinary meaning of a claim term is not as simple as looking up the term in a general-purpose dictionary (or even a technical dictionary). *Id.* at 1322 (noting that "discrepancies between the patent" and "technical dictionaries or treatises" are "apt to be common"). "[E]levating the dictionary" to the primary source, as the district court did here, "focuses the inquiry on the abstract meaning of words rather than on the meaning of claim terms within the context of the patent." *Id.* at 1321. As *Phillips* explained, such "heavy reliance on the dictionary divorced from the intrinsic evidence risks

transforming the meaning of the claim term to the artisan into the meaning of the term in the abstract, out of its particular context, which is the specification." *Id.*

For example, in *Advanced Fiber*, this Court reversed a district court's construction of "perforated" because the district court relied on dictionary definitions of "perforate" and "perforation" that were too narrow in light of the specification. *Advanced Fiber Techs. (AFT) Trust v. J & L Fiber Servs., Inc.*, 674 F.3d 1365, 1374-75 (Fed. Cir. 2012). Although the dictionary definitions provided that perforation is made through piercing or puncturing, the claims and specification required merely a plurality of openings. *Id.* This Court held that the district court erred in relying "solely on dictionary definitions" that "contradicted the patent's specification." *Id.*

Here, as in *Advanced Fiber*, the district court construed "identifying" and "recognizing" in a vacuum, divorced from the patent description. In the summary-judgment briefing, it became clear that there was a dispute over the meaning of "identifying" and "recognizing" that had not been resolved during initial claim construction. Specifically, the parties disagreed about whether those terms require a final, unambiguous identification. Nuance thus asked the district court to construe the terms.

The district court ordered the parties "to arrive at a mutually agreed upon -- quote, unquote -- 'ordinary meaning'" of "identifying" and "recognizing." A1368.

Expressing reluctance to construe the terms, the district court stated that if the parties could not come to an agreement, the court would use a definition from a general-purpose dictionary: "I think it's too late to do a construction. I think it's unnecessary. I'd like to avoid it. And if the parties don't agree, then *I will just either use a Black's Law Dictionary definition, or some other definition*, or just tell the jury to use its ordinary meaning." A1368-A1369 (emphasis added). When the parties could not agree, the district court simply adopted one of the Merriam-Webster definitions without indicating that it considered the specification at all. That approach cannot be squared with this Court's well-settled rules governing claim construction.

Contrary to the reasons given by the district court, it was not "too late" for the court to resolve the dispute over the meaning of claim terms shortly before trial. Indeed, it was the court's duty to do so. *O2 Micro Int'l Ltd. v. Beyond Innovation Tech. Co.*, 521 F.3d 1351, 1360 (Fed. Cir. 2008) ("When the parties raise an actual dispute regarding the proper scope of these claims, the court, not the jury, must resolve that dispute."). Nor does the district court's reluctance to resolve a late-arising claim-construction issue justify using a general-purpose dictionary definition in lieu of applying well-established principles of claim construction. *See id.*

To be sure, district courts may consult dictionaries during claim construction "'to better understand the underlying technology,'" and they "'may also rely on dictionary definitions when construing claim terms.'" *Phillips*, 415 F.3d at 1322 (quoting *Vitronics*, 90 F.3d at 1584 n.6). But the district court must "ensure that any reliance on dictionaries accords with the intrinsic evidence." *Free Motion Fitness, Inc. v. Cybex Int'l, Inc.*, 423 F.3d 1343, 1348 (Fed. Cir. 2005). Where, as here, the dictionary definition "'contradict[s] any definition found in or ascertained by a reading of the patent documents,'" the dictionary definition must give way. *Phillips*, 415 F.3d at 1322-23 (quoting *Vitronics*, 90 F.3d at 1584 n.6); *see Renishaw PLC v. Marposs Societa' per Azioni*, 158 F.3d 1243, 1250 (Fed. Cir. 1998) ("[A] common meaning, such as one expressed in a relevant dictionary, that flies in the face of the patent disclosure is undeserving of fealty."). "[A] general-usage dictionary cannot overcome credible art-specific evidence of the meaning" of a claim term. *Vanderlande Indus. Nederland BV v. ITC*, 366 F.3d 1311, 1321 (Fed. Cir. 2004). In fact, if the meaning of a term is apparent based on the intrinsic evidence, it is "improper to rely on extrinsic evidence" at all. *Vitronics*, 90 F.3d at 1583.

### B. Properly Construed, “Identifying” And “Recognizing” Encompass Both Final And Tentative Identifications

The district court’s improper claim-construction analysis led it to reach an incorrect construction. Properly construed, “identifying” and “recognizing” as used in the ’342 patent include non-final identifications.

#### *1. Intrinsic evidence expressly provides that identification may be tentative or equivocal*

The ’342 patent’s specification expressly teaches that an unknown character may be identified or recognized by narrowing the potential matches for the character to a plurality of possible matches—that is, “identifying” and “recognizing” includes ambiguous identifications. The specification explains that after a character is identified using feature analysis, context analysis may be performed to resolve ambiguities in the identification. According to the specification, feature analysis “*identifies* characters by their shape. However the shape of the character alone may not be deterministic of what the character is.” A59(col.25:2-4) (emphasis added). Thus, “[c]ontext analysis is performed . . . to attempt to resolve ambiguities.” A59(col.25:10-11). “A process termed context analysis is employed to examine the relative sizes and positions of the shapes *recognized* during the character recognition process to divide the text into words and to resolve ambiguity of shape . . . .” A48(col.4:30-33); *see* A59 (col.26:12-14) (“After preparing the line for context analysis, block 1501, a first pass is made

through each character on the line, block 1502, to attempt to resolve ambiguities."). In a preferred embodiment, the context analysis feature uses a database that has "information regarding the relative size of characters which are normally *ambiguous when identified* by shape alone." A59(col.25:20-22) (emphasis added).

The specification also describes a preferred embodiment in which feature analysis involves using isit routines, which "distinguish characters based on their shape." A56(col.20:5-6). The specification teaches that "characters sharing the same topography are *recognized* by a single isit." A56(col.20:6-7) (emphasis added). In other words, identifying or recognizing a character includes identifying that the unknown character belongs to a set of characters with the same basic shape. "For example, the letter lower case 'p' and the letter upper case 'P' are *recognized* by the same isit." A56(col.20:7-9) (emphasis added). "The letters lower case 'u' and upper case 'U', lower case 's' and upper case 'S', lower case 'o', upper case 'O' and zero '0', etc., are other examples of characters with the same or similar topography which would be *recognized* by the same isits." A56(col.20:9-13) (emphasis added).

The specification's description of the outputs from isits states explicitly that identification includes equivocal and unequivocal identifications. If an isit makes an unambiguous identification, it outputs "an ASCII code for a particular

character." A56(col.20:19). "An output of the ASCII code for a particular character indicates the character's identification by the isit routine is unequivocal." A56(col.20:22-24). But if an ambiguous identification is made, the isit outputs "a code indicating the character is *recognized as belonging to a particular class of characters*." A56(col.20:19-21) (emphasis added). Only if the isit cannot identify a class of characters to which the unknown character belongs does the isit output "a reject code indicating the character is not recognized." A56(col.20:21-22).

A person of skill in the art would read these express, consistent statements throughout the specification as implying a definition of "identifying" and "recognizing" that includes equivocal identifications. *Bell Atl. Network Servs., Inc. v. Covad Commc'ns Grp., Inc.*, 262 F.3d 1258, 1271 (Fed. Cir. 2001). A skilled artisan would not understand the claims to have the district court's construction because, under that construction, these examples from the preferred embodiment would not meet the claims. A "claim construction that excludes a preferred embodiment . . . 'is rarely, if ever, correct.'" *SanDisk Corp. v. Memorex Prods., Inc.*, 415 F.3d 1278, 1285 (Fed. Cir. 2005) (quoting *Vitronics*, 90 F.3d at 1583). "[I]t is unlikely that an inventor would define the invention in a way that excluded the preferred embodiment, or that persons of skill in this field would read the specification in such a way." *Hoechst Celanese Corp. v. BP Chems. Ltd.*, 78 F.3d 1575, 1581 (Fed. Cir. 1996).

Claim language in claim 7 of the ’489 patent, which was asserted at trial, confirms that “identifying” can result in ambiguous identification. The ’489 and ’342 patents are in the same family and share a common specification, and both patents were construed together. Claim 7 of the ’489 patent depends from claim 2, which contains the step of “identifying said unknown character” followed by the step of “resolving ambiguities for a character.” A103(col.29:3, col.29:12). Thus, in claim 7, it is explicit that “identifying” can result in an ambiguous identification. That same meaning applies to “identifying” in the ’342 patent. *Omega Eng’g, Inc. v. Raytek Corp.*, 334 F.3d 1314, 1334 (Fed. Cir. 2003) (“[U]nless otherwise compelled . . . the same claim term in the same patent or related patents carries the same construed meaning.”).

Thus, in light of the intrinsic evidence, one of ordinary skill in the art would understand that “identifying” and “recognizing” in the context of this art includes ambiguous identifications.

#### ***2. ABBYY’s own statements confirm that, to a person skilled in the art, “identification” includes equivocal identification***

That this is the meaning of “identifying” to a skilled artisan is confirmed by the initial claim-construction proceedings. Before ABBYY reversed itself in search of a non-infringement defense, the parties agreed that the ordinary meaning of “identifying” may include ambiguous identifications.

During the initial claim-construction proceedings, ABBYY took the position that "identifying" is limited to "template matching, feature analysis, or both." A543. Nuance contended that "identifying" should have its ordinary meaning to a person skilled in the art. A519-A521. While the parties disagreed as to whether "identifying" is limited to specific types of character recognition, both sides agreed that the ordinary meaning of "identifying" in this field of art includes equivocal identifications. Specifically, in its claim-construction brief, ABBYY explained that the "product of [feature] analysis is *identification* of the unknown character." A537 (emphasis added). ABBYY went on to explain that "[m]any times *an identified character is still ambiguous*. For example, is it an 's' or an 'S'? Each has similar features. Contextual analysis resolves these ambiguities through a number of different ways." A537 (emphasis added).

Later, when the disagreement arose over whether "identifying" requires unambiguous identification, ABBYY pointed to Nuance's earlier position that "identifying" should be given its ordinary meaning. A1438. ABBYY argued that because Nuance had proposed an ordinary-meaning construction, the district court could use a dictionary definition. A1438. But the ordinary meaning that Nuance proposed was the ordinary meaning to one of skill in the art, as this Court's decisions require. *Phillips*, 415 F.3d at 1312-13. ABBYY's statement in its claim-construction brief that an identified character may still be ambiguous is consistent

with how the patent uses "identifying" and "recognizing." That statement, not a general-purpose dictionary definition, accurately reflects how a person skilled in the art would understand "identifying" and "recognizing" in the context of the '342 patent.

#### ***3. Extrinsic evidence confirms that "identifying" and "recognizing" can include equivocal identification***

ABBYY's expert Dr. Doermann confirmed in his deposition that, to a person skilled in the art, "identifying" does not mean reaching a final identification. Dr. Doermann agreed that the purpose of context analysis "is to resolve ambiguities that resulted or could have resulted from the step of *identifying* said unknown character." A1078 (emphasis added). He also agreed that a person of ordinary skill in the art "would understand that 'identifying' does not refer to the final step in the method for recognizing a character." A1075-A1076. Dr. Doermann confirmed that in a publication he co-authored, he used "identify" to refer to a process that could result in a plurality of entries that reasonably correspond to a desired result. A1081.

Other art in the field of OCR further confirms that, to a skilled artisan, "identifying" and "recognizing" includes ambiguous identifications. One such piece of art is ABBYY's own patent, U.S. Patent No. 7,769,235 ("'235 patent"). ABBYY's '235 patent explicitly states that "identification" can result in "a plurality of possible characters" and that context analysis "refines the results of the

recognition process." A1035(col.2:42-47). ABBYY's patent notes that after the "image is recognized, the entire plurality of symbols . . . is sent to context analysis . . . for final identification." A1036(col.4:13-16). The patent also describes how context analysis is used to "examine the correctness of recognition" and to "reduce the plurality of possible characters." A1035(col.2:26-28). Likewise, U.S. Patent No. 4,058,795, which was cited as prior art in ABBYY's invalidity contentions and expert report, teaches making a "final identification of the unknown character" from "provisional identifications." A1048(col.1:60-62); *see* A1040(Abstract), A1048-A1055(col.1:51-60, col.5:29-32, col.5:47-49, col.6:26-29, col.7:8-11, col.11:53-58, col.14:51-54, col.15:54-56).

In light of the intrinsic and extrinsic evidence, the district court should have construed "identifying" and "recognizing" as including both final and non-final identifications.

### C. Nuance Is Entitled To A New Trial On All Its Claims With Respect To The '342 Patent Under The Correct Construction

The case should be remanded for a new trial on the asserted claims of the '342 patent under the correct construction of "identifying" and "recognizing." Where this Court "adopt[s] a new claim construction on appeal" and it is not "clear from the record that the accused device does or does not infringe, a remand is warranted for a determination of infringement under the correct claim construction." *Praxair, Inc. v. ATMI, Inc.*, 543 F.3d 1306, 1324 (Fed. Cir. 2008).

It is beyond dispute that the construction of “identifying” and “recognizing” may have led to the jury’s non-infringement verdict. ABBYY’s non-infringement theory at trial centered on whether FineReader’s Omnifont classifiers identify or recognize characters. A2095-A2103; A3705-A3710. ABBYY’s expert, Dr. Doermann, testified that under the district court’s construction, identifying a character requires determining the particular computer code (the ASCII code) for the unknown character—that is, making a final identification. A2096. He then asserted that because FineReader’s Omnifont classifiers generate up to eight variants, they do not perform the claim 4 limitation reciting “identifying said unknown character with said feature analysis process.” A60(col.28:1-2); A2098. Offering the same explanation, Dr. Doermann also testified that FineReader “doesn’t recognize” within the meaning of the limitation of claim 13 that recites “generating a character template, if said character is recognized using said second recognition process.” A61(col.29:27-28); *see* A2101.

Indeed, ABBYY acknowledged below that the reason it prevailed may have been that the jury rejected the testimony of Nuance’s expert “as inconsistent with the Court’s construction of ‘identify.’” A3706; *see* A3708 (arguing that Dr. Lopresti’s “definition of ‘identify’ was reason enough for the jury to reject Dr. Lopresti’s testimony”). Under a different construction, the jury could not have rejected that testimony for that reason.

The judgment of non-infringement of the asserted claims of the ’342 patent should therefore be set aside, and the case should be remanded for a new trial on all issues with respect to the ’342 patent, including direct and indirect infringement, willfulness, and damages. *Praxair*, 543 F.3d at 1324; *Pumphrey v. K.W. Thompson Tool Co.*, 62 F.3d 1128, 1133-34 (9th Cir. 1995) (new trial on one issue requires new trial on all related issues).

## II. THE DISTRICT COURT SHOULD NOT HAVE PRECLUDED NUANCE FROM PURSUING ITS CLAIMS ON THE FIVE PATENTS THAT WERE NOT SELECTED FOR THE FIRST TRIAL

The district court erroneously entered judgment against Nuance on five patents that were never adjudicated. That judgment should be vacated, and the case remanded so Nuance can pursue its untried claims.

### A. Nuance Has A Right To Have Its Infringement Claims Adjudicated On All Its Asserted Patents

As patentee, Nuance has a right to have its claims of infringement adjudicated on all eight patents on which it sued. Congress decreed that “[a] patentee shall have remedy by civil action for infringement of his patent.” 35 U.S.C. § 281. “By statutory and common law, each patent establishes an independent and distinct property right.” *Kearns v. General Motors Corp.*, 94 F.3d 1553, 1555 (Fed. Cir. 1996). Thus, “[e]ach patent asserted raises an independent and distinct cause of action.” *Id*.

The district court has an obligation to adjudicate distinct causes of action that are brought before it. *See Fin Control Sys. Pty, Ltd. v. OAM, Inc.*, 265 F.3d 1311, 1321 (Fed. Cir. 2001) ("[B]ecause OAM had raised the alleged invalidity and unenforceability of the '359 patent in its counterclaim, the court was obligated to rule on these matters as a prerequisite to entering judgment in the case." (citing *Cardinal Chem. Co. v. Morton Int'l, Inc.*, 508 U.S. 83, 93-94 (1993)). A district court cannot force a plaintiff to relinquish a cause of action; once a plaintiff brings suit to enforce its rights, "[t]here is no duty . . . to settle cases, or to reduce one's claims." *Del Rio v. Northern Blower Co.*, 574 F.2d 23, 26 (1st Cir. 1978). "Unless his claim is frivolous, a party is entitled to assert it, and to whatever judicial time is required to try it." *Id.*; *see La Buy v. Howes Leather Co.*, 352 U.S. 249, 258 (1957) ("'Litigants are entitled to a trial by the court, in every suit, save where exceptional circumstances are shown.'" (quoting *Adventures in Good Eating, Inc. v. Best Places to Eat, Inc.*, 131 F.2d 809, 815 (7th Cir. 1942))).

Thus, while a district court has discretion over issues of case management, there are definite limits on the district court's ability to enter judgment on a claim without actually deciding its merits. In particular, the party asserting the claim must have the opportunity to present evidence and argument before disposition of its claim. *See, e.g.*, Fed. R. Civ. P. 56(f). Thus, where the district court enters judgment "on a ground for which a full and fair opportunity to proffer material

evidence was not offered, this court must vacate and remand." *Massey v. Del Labs., Inc.*, 118 F.3d 1568, 1573 (Fed. Cir. 1997); *see Fin Control*, 265 F.3d at 1321.

These protections are essential not only to protecting the orderly administration of justice, but also to protecting a patentee's due process rights. It is well established that a patent is a property right protected by the Due Process Clause. *Florida Prepaid Postsecondary Educ. Expense Bd. v. College Sav. Bank*, 527 U.S. 627, 642 (1999); *Patlex Corp. v. Mossinghoff*, 758 F.2d 594, 599 (Fed. Cir. 1985). Therefore, as with any property right, a patentee must be provided "the opportunity to be heard 'at a meaningful time and in a meaningful manner'" before being deprived of its claim for infringement of its patent rights. *Mathews v. Eldridge*, 424 U.S. 319, 333 (1976).

These principles are reinforced by *In re Katz*. Unlike here, the district court there ensured that the patentee had a full and fair opportunity to be heard on all its claims. *In re Katz Interactive Call Processing Patent Litig.*, 639 F.3d 1303, 1309 (Fed. Cir. 2011)). In that case, the district court initially instructed Katz to select 64 claims to pursue at trial—eight claims for each group of asserted patents relating to a particular technology. *Id.* The district court allowed Katz to add new claims if Katz showed that the claims raised separate legal issues that were not duplicative of the issues raised by already-selected claims. *Id.* Katz never

attempted to make that showing, and when the district court entered summary judgment on the claims that Katz had selected for trial, the court also did so on the unselected claims. *Id.* at 1309-10, 1313. This Court approved of that procedure as affording Katz its due process rights, explaining that "it was both efficient and fair to require Katz to identify those unasserted claims that, in Katz's view, raised separate legal issues from those raised by the asserted claims." *Id.* at 1312.

The Court emphasized, however, that if Katz had shown "that some of its unselected claims presented unique issues as to liability or damages" and that, "notwithstanding such a showing, the district court had refused to permit Katz to add those specified claims, that decision would be subject to review *and reversal*." *Id.* at 1312-13 (emphasis added). That is the outcome warranted here.

### B. The District Court Erroneously Entered Judgment Of Non-Infringement On Five Nuance Patents That Were Never Adjudicated

Here, the district court's decision to treat the jury's verdict of non-infringement on the three patents selected for trial as also applying to Nuance's five unselected patents was erroneous and should be reversed.

Unlike in *Katz*, the patents that were tried were not representative of the unselected patents. Nuance's unselected '094, '404, and '009 patents are completely unrelated to the tried patents. In addition, although Nuance's '053 and '983 patents are related to the tried '342 and '489 patents, the asserted claims focus

on different aspects of the inventive OCR system. Indeed, ABBYY asserted early on that Nuance's "eight patents cover[ed] different technologies." A695. For that reason, rather than seeking to treat claims as duplicative, ABBYY initially asked the district court to *sever* them: "there is no advantage to litigating these completely independent causes of action in the same case involving the OCR-related patents," i.e., the '342, '489, and '009 patents. A695. This is alone reason to reverse, as *Katz* emphasized that where the district court does not allow a patentee to pursue claims that present unique issues, the decision will "be subject to review and reversal." 639 F.3d at 1312-13.

Nor did Nuance ever abandon its five unselected patents or stipulate that the judgment on the patents selected for trial would apply to all patents. Although Nuance complied with the district court's orders to select certain claims for the first trial, at every opportunity Nuance made clear that it was not agreeing to forgo adjudication of its unselected patents. For example, Nuance stated that although it would "narrow the case for expert discovery and trial," it was "reserving its rights on all other asserted patents and claims." A726. Nuance stated that it was agreeing only to "postpone resolution of its infringement case" as to the remaining patents. A726. Nuance expressly "reserve[d] its right to reassert them against ABBYY and/or Lexmark at a later time in this suit or a future suit(s)." A727. In a joint case-management report, Nuance emphasized that it "has no intention or

desire to drop its claims on the other five patents that it believes are being infringed by ABBYY and Lexmark based on the resolution of the three patents picked . . . to be construed by the Court first." A665-A666. Nuance thus repeatedly told the district court that it understood the claim-selection process to affect only *when* its claims would be litigated, not *whether* its claims would be litigated, and the district court never suggested otherwise. To the contrary, the district court assured Nuance that it eventually would have its day in court on all of its claims. A435, A449; A477.

Nevertheless, after the first trial, the district court treated the jury's verdict as a judgment of non-infringement on the patents that were never tried—despite that the patents are directed to different technologies, no representative claims or issues were before the jury, and Nuance never agreed to this procedure or even had notice that it would be employed. The district court had no discretion to treat Nuance's claims in this manner. To the contrary, the district court had an obligation to allow Nuance to present evidence and argument on all of its patents, as each is an independent cause of action. The district court's failure to do so exceeded its case-management authority, and that alone is reason to vacate the judgment and remand for a determination on the merits of the untried claims. Although this Court need not reach the constitutional question to reverse, the

district court's actions also deprived Nuance of due process, and that likewise is an independent ground for vacatur and remand.

### C. None Of The District Court's Reasons Justify Its Entry Of Judgment On Nuance's Patents That Were Never Adjudicated

In entering judgment on Nuance's untried patents, the district court offered several reasons, none of which withstands scrutiny.

The district court stated that "there was never any mention that there would be serial trials." A23. But Nuance repeatedly cautioned the district court that its approach may result in two trials. At an early case-management conference, Nuance raised the possibility that "*two trials*" would be needed. A421 (emphasis added). Nuance also made clear it was agreeing simply to put certain patents "on the side burner for now" so that those patents "are not going to be in the *first* trial." A418-A420 (emphasis added). Later, in the joint case-management report, Nuance again noted the possibility that this course could lead to a second trial: "If the case is unable to be resolved before trial, it is only fair that Nuance—not any of the Defendants—be able to select the patents for the *first (and hopefully only) trial in this case*." A666 (emphasis added).

The district court suggested that Nuance "selected its best and strongest patents for trial." A23. But Nuance's rights are not limited to the claims it may at one point have regarded as its strongest. *Katz*, 639 F.3d at 1312-13. What matters is whether the claims are distinct, *id.*, and here all the parties agreed that the

patents covered distinct technologies. Regardless of the perceived strength of a cause of action, if the district court enters judgment on it without providing "a full and fair opportunity to proffer material evidence . . . , this court must vacate and remand." *Massey*, 118 F.3d at 1573.

Finally, the district court erred in faulting Nuance for not raising the issue of its untried claims in its post-trial motions for judgment as a matter of law or, in the alternative, for a new trial. A23-A24. Nuance had no notice that it needed to raise these claims in its post-trial motions seeking to set aside the jury's verdict. The jury rendered a verdict only as to the claims that were actually submitted to the jury. A3631-A3636. Likewise, the judgment (which was entered on the same day as the jury's verdict) stated merely: "*Pursuant to the jury's verdict*, . . . JUDGMENT is HEREBY ENTERED in favor of Defendants and against Plaintiff." A17 (emphasis added); *see* Fed. R. Civ. P. 58(b)(1). The scope of that judgment was necessarily limited only "to the issues that were actually litigated." *Alcon Research Ltd. v. Barr Labs., Inc.*, 745 F.3d 1180, 1193 (Fed. Cir. 2014); *see Tol-O-Matic, Inc. v. Proma Produkt-Und Mktg. Gesellschaft m.b.H.*, 945 F.2d 1546, 1554-55 (Fed. Cir. 1991), *abrogated on other grounds by Markman*, 52 F.3d 967. In any event, a motion for judgment as a matter of law can be raised only as to an issue on which "a party has been fully heard . . . during a jury trial." Fed. R. Civ. P. 50(a)(1). Because the untried claims were not within the scope of the

jury's verdict or the district court's judgment entered pursuant to that verdict, there was no reason for Nuance to raise those claims in its motion seeking to set aside the verdict.

* * *

In sum, to the extent the district court thought it necessary to limit the claims and patents that Nuance could adjudicate, *Katz* offered a roadmap. The district court strayed far from that roadmap and in so doing both abused its discretion and deprived Nuance of due process. Either error requires setting aside the judgment with respect to Nuance's as-yet untried patents and remanding for further proceedings on those claims.

## CONCLUSION

The district court's construction of "identifying" and "recognizing" should be set aside, and a new trial should be ordered on all issues with respect to the '342 patent. The district court's judgment as to the '053, '983, '009, '404, and '094 patents also should be vacated and the case remanded for further proceedings.

Respectfully submitted,

Dated: November 14, 2014

/s/ Deanne E. Maynard

DEANNE E. MAYNARD
MARC A. HEARRON
JESSICA E. PALMER
MORRISON & FOERSTER LLP
2000 Pennsylvania Avenue, NW
Washington, DC 20006
Telephone: (202) 887-8740
DMaynard@mofo.com

MICHAEL A. JACOBS
MORRISON & FOERSTER LLP
425 Market Street
San Francisco, CA 94105
Telephone: (415) 268-7000

*Counsel for Plaintiff-Appellant Nuance Communications, Inc.*

# ADDENDUM

IN THE UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA

NUANCE COMMUNICATIONS, INC. A Delaware corporation,

Plaintiff,

v.

ABBYY SOFTWARE HOUSE, INC., a California corporation, and LEXMARK INTERNATIONAL, INC., a Delaware corporation,

Defendants.
_______________________________________/

No. C 08-02912 JSW

**CLAIM CONSTRUCTION ORDER**

The Court has been presented with a technology tutorial and briefing leading up to a hearing pursuant to *Markman v. Westview Instruments, Inc.*, 517 U.S. 370 (1996). This Order construes the eleven claim terms selected by the parties, which appear in several of the patents at issue in this case, United States Patent No. 6,038,342 ("the '342 Patent") called 'Optical Character Recognition and Apparatus," United States Patent No. 5,261,009 ("the '009 Patent") called 'Means for Resolving Ambiguities in Text Passed Upon Character Context," and United States Patent No. 381,489 ("the '489 Patent") called "Optical Recognition Method and Apparatus."

**BACKGROUND**

Nuance Communications, Inc. ("Nuance") contends that the remaining defendants, Abbyy Software House and Abbyy USA Software House, Inc. ("Abbyy") and Lexmark International, Inc. (collectively "Abbyy/Lexmark") infringe nine of Nuance's imaging patents.

Of those nine, five relate directly to Optical Character Recognition ("OCR") and are the subject of the present claim construction process. OCR refers to the translation of an image containing text to an editable format.

The first commercially available OCR systems were "single-font" systems, capable of recognizing only one character font. The second class of OCR systems, referred to as omnifont systems, was able to recognize a much larger number of fonts in a wide range of point size and spacing, but performed slowly due to the unpredictability and greater variation of input. According to Nuance's OmniPage series of patents, the intention was to introduce significant improvements to the OCR technology, thereby allowing omnifont systems to operate as quickly as single-font counterparts, which retaining their flexibility.

The Court shall address additional facts as necessary in the remainder of this Order.

**ANALYSIS**

**A. Legal Standard.**

"It is a bedrock principle of patent law that the claims of a patent define the invention to which the patentee is entitled the right to exclude." *Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1115 (Fed. Cir. 2004). The interpretation of the scope and meaning of disputed terms in patent claims is a question of law and exclusively within the province of a court to decide. *Markman*, 517 U.S. at 372. The inquiry into the meaning of the claim terms is "an objective one." *Innova/Pure Water*, 381 F.3d at 1116. As a result, when a court construes disputed terms, it "looks to those sources available to the public that show what a person of skill in the art would have understood the disputed claim language to mean." *Id.* In most cases, a court's analysis will focus on three sources: the claims, the specification, and the prosecution history. *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 979 (Fed. Cir. 1995) (en banc), *aff'd*, 517 U.S. 370 (1996). However, on occasion, it is appropriate to rely on extrinsic evidence regarding the relevant scientific principles, the meaning of technical terms, and the state of the art at the time at the time the patent issued. *Id.* at 979-981.

**United States District Court**
For the Northern District of California

The starting point of the claim construction analysis is an examination of the specific claim language. A court's "claim construction analysis must begin and remain centered on the claim language itself, for that is the language that the patentee has chosen to particularly point out and distinctly claim the subject matter which the patentee regards as his invention." *Innova/Pure Water*, 381 F.3d at 1116 (internal quotations and citations omitted). Indeed, in the absence of an express intent to impart a novel meaning to a term, an inventor's chosen language is given its ordinary meaning. *York Prods., Inc. v. Cent. Tractor Farm & Family Center*, 99 F.3d 1568, 1572 (Fed. Cir. 1996). Thus, "[c]laim language generally carries the ordinary meaning of the words in their normal usage in the field of the invention." *Invitrogen Corp. v. Biocrest Mfg., L.P.*, 327 F.3d 1364, 1367 (Fed. Cir. 2003); *see also Renishaw v. Marposs Societa' per Azioni*, 158 F.3d 1243, 1248 (Fed. Cir. 1998) (recognizing that "the claims define the scope of the right to exclude; the claim construction inquiry, therefore, begins and ends in all cases with the actual words of the claim"). A court's final construction, therefore, must accord with the words chosen by the patentee to mete out the boundaries of the claimed invention.

The court should also look to intrinsic evidence, including the written description, the drawings, and the prosecution history, if included in the record, to provide context and clarification regarding the intended meaning of the claim terms. *Teleflex, Inc. v. Ficosa N. Am. Corp.*, 299 F.3d 1313, 1324-25 (Fed. Cir. 2002). The claims do not stand alone. Rather, "they are part of 'a fully integrated written instrument.'" *Phillips v. AWH Corp.*, 415 F.3d 1303, 1315 (Fed. Cir. 2005) (en banc) (quoting *Markman*, 52 F.3d at 978). The specification "may act as a sort of dictionary, which explains the invention and may define terms used in the claims." *Markman*, 52 F.3d at 979. The specification also can indicate whether the patentee intended to limit the scope of a claim, despite the use of seemingly broad claim language. *SciMed Life Sys., Inc. v. Advanced Cardiovascular Sys., Inc.*, 242 F.3d 1337, 1341 (Fed. Cir. 2001) (recognizing that when the specification "makes clear that the invention does not include a particular feature, that feature is deemed to be outside the reach of the claims of the patent, even though the

language of the claims, read without reference to the specification, might be considered broad enough to encompass the feature in question”).

Intent to limit the claims can be demonstrated in a number of ways. For example, if the patentee “acted as his own lexicographer,” and clearly and precisely “set forth a definition of the disputed claim term in either the specification or prosecution history,” a court will defer to that definition. *CCS Fitness, Inc. v. Brunswick Corp.*, 288 F.3d 1359, 1366 (Fed. Cir. 2002). In order to so limit the claims, “the patent applicant [must] set out the different meaning in the specification in a manner sufficient to give one of ordinary skill in the art notice of the change from ordinary meaning.” *Innova/Pure Water*, 381 F.3d at 1117. In addition, a court will adopt an alternative meaning of a term “if the intrinsic evidence shows that the patentee distinguished that term from prior art on the basis of a particular embodiment, expressly disclaimed subject matter, or described a particular embodiment as important to the invention.” *CCS Fitness*, 288 F.3d at 1367. For example the presumption of ordinary meaning will give way where the “inventor has disavowed or disclaimed scope of coverage, by using words or expressions of manifest exclusion or restriction, representing clear disavowal of claim scope.” *Gemstar-TV Guide Int’l Inc. v. ITC*, 383 F.3d 1352, 1364 (Fed. Cir. 2004). Likewise, the specification may be used to resolve ambiguity “where the ordinary and accustomed meaning of the words used in the claims lack sufficient clarity to permit the scope of the claim to be ascertained from the words alone.” *Teleflex*, 299 F.3d at 1325.

However, limitations from the specification (such as from the preferred embodiment) may not be read into the claims, absent the inventor’s express intention to the contrary. *Id.* at 1326; *see also CCS Fitness*, 288 F.3d at 1366 (“[A] patentee need not ‘describe in the specification every conceivable and possible future embodiment of his invention.’”) (quoting *Rexnord Corp. v. Laitram Corp.*, 274 F.3d 1336, 1344 (Fed. Cir. 2001)). To protect against this result, a court’s focus should remain on understanding how a person of ordinary skill in the art would understand the claim terms. *Phillips*, 415 F.3d at 1323.

**United States District Court**
For the Northern District of California

If the analysis of the intrinsic evidence fails to resolve any ambiguity in the claim language, a court then may turn to extrinsic evidence, such as expert declarations and testimony from the inventors. *Intel Corp. v. VIA Techs., Inc.*, 319 F.3d 1357, 1367 (Fed. Cir. 2003) ("When an analysis of *intrinsic* evidence resolves any ambiguity in a disputed claim term, it is improper to rely on extrinsic evidence to contradict the meaning so ascertained.") (emphasis in original). When considering extrinsic evidence, a court should take care not to use it to vary or contradict the claim terms. Rather, extrinsic evidence is relied upon more appropriately to assist in determining the meaning or scope of technical terms in the claims. *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1583-84 (Fed. Cir. 1996).

Dictionaries also may play a role in the determination of the ordinary and customary meaning of a claim term. In *Phillips*, the Federal Circuit reiterated that "[d]ictionaries or comparable sources are often useful to assist in understanding the commonly understood meanings of words...." *Phillips*, 415 F.3d at 1322. The *Phillips* court, however, also admonished that district courts should be careful not to allow dictionary definitions to supplant the inventor's understanding of the claimed subject matter. "The main problem with elevating the dictionary to ... prominence is that it focuses the inquiry on the abstract meaning of the words rather than on the meaning of claim terms within in the context of the patent." *Id.* at 1321. Accordingly, dictionaries necessarily must play a role subordinate to the intrinsic evidence.

In addition, a court has the discretion to rely upon prior art, whether or not cited in the specification or the file history, but only when the meaning of the disputed terms cannot be ascertained from a careful reading of the public record. *Vitronics*, 90 F.3d at 1584. Referring to prior art may make it unnecessary to rely upon expert testimony, because prior art may be indicative of what those skilled in the art generally understood certain terms to mean. *Id.*

**B. Claim Construction.**

**1. "Template"**

Claim 1 of the '342 Patent recites a "method of optically recognizing characters on a medium and storing a template of said recognized characters in a template cache for recognition of subsequent characters on said medium...." (27:29-32.) The word "template" also appears in Claims 4, 6, 7, 13, 14, and 18.

Abbyy/Lexmark argue that the term "template" must be construed to mean "A data structure comprising a header field comprising offset pointers to the original pattern of the character recognized through feature analysis, a 'must-be-off' pattern derived from said original pattern, and a 'must-be-on' pattern derived from said original pattern." (Parties' Amended Final Joint Claim Construction and Pre-hearing Statement ("Statement"), Ex. 1.) Nuance, on the other hand, argues that only the word "template" should be construed to mean "A representation of the patterns, shapes, or images of a character." (*Id.*)

The key distinction between the two proffered constructions is whether the embodiment of the "must-be-on" and "must-be-off" patterns should be read into the construction of the claim term. In fact, in Claim 2, 3, 8, and 15, the patent actually reserves the dependent claims to cover the specific embodiment in which the template may consist of the "must-be-on" and "must-be-off" patterns. Abbyy/Lexmark's proposed construction would render the dependent claims superfluous.

In general, the doctrine of claim differentiation recognizes "that different words or phrases used in separate claims are presumed to indicate that the claims have different meanings and scope." *Andersen Corp. v. Fiber Composites, LLC*, 474 F.3d 1361, 1369 (Fed. Cir. 2007) (quoting *Karlin Tech. Inc. v. Surgical Dynamics, Inc.*, 177 F.3d 968, 971-72 (Fed. Cir. 1999)). Thus, there is a presumption that "[t]o the extent the absence of such difference in meaning and scope would make a claim superfluous, ... the difference between claims is significant." *Id.* (quoting *Tandon Corp. v. U.S. Int'l Trade Comm'n*, 831 F.2d 1017, 1023 (Fed. Cir. 1987)).

That presumption may be overcome, however, by the written description of the patent or its prosecution history. *Id.*

The Court finds that the general term "template" should be construed using the broad construction proffered by Nuance, in which any particular template may – or may not – include the "must-be-on" and "must-be-off" patterns. The Court finds no reason to overcome the presumption of claim differentiation or to render the dependent claims in the patents entirely superfluous or redundant. The specification permits for an allowance that characters may be within predetermined tolerances, but does not require that they be either in the must-be-off or must-be-on patterns.

Accordingly, the Court construes the term "template" to mean: "A representation of the patterns, shapes, or images of a character."

**2. "Character"**

Claim 1 of the '342 Patent recites a "method for optically recognizing characters on a medium and storing a template of said recognized characters in a template cache for recognition of subsequent characters on said medium...." (27:29-32.) The word "character" also appears in Claims 4, 6, 7, 13, 14 (as dependent on 13), and 18. The word "character" also appears in Claim 7 of the '489 Patent and Claim 22 of the '009 Patent.

Abbyy/Lexmark argue that the term "character" should be construed to mean "A character is a single, individual symbol from a known set of symbols, not joined with any other symbol, identifiable as such by a segmentation process. Examples of characters are letters, numerals and special numbers such as commas, quotation marks and semicolons." (Statement, Ex. 1.) Nuance, on the other hand, argues that the term "character" should be construed to mean "One or a group of adjacent letters, digits, or other symbols." (*Id.*)

The key distinction between the two proffered constructions is that Abbyy/Lexmark attempts to narrow the construction to refer to only a single, identified letter, not, as Nuance proffers, a possible multiple symbol character. Nuance contends that a term singular may be construed to connote both a singular and plural form of the word. *See Baldwin Graphic Sys.*,

*Inc. v. Siebert, Inc.*, 512 F.3d 1338, 1342-43 (Fed. Cir. 2008) ("That 'a' or 'an' can mean 'one or more' is best described as a rule, rather than merely a presumption or even a convention. The exceptions to this rule are extremely limited: a patentee must 'evince[] a clear intent' to limit 'a' or 'an' to 'one.'"); *KCJ Corp. v. Kinetic Concepts, Inc.*, 223 F.3d 1351, 1356 (Fed. Cir. 2000) ("Unless the claim is specific as to the number of elements, the article 'a' receives a singular interpretation only in rare circumstances when the patentee evinces a clear intent to so limit the article."). Further, Nuance argues, the inventors recognized that the segmentation process may sometimes yield a multi-symbol character, such as 'ite' and explicitly accounts for this situation in the patent. (*See* '342 Patent at 24:62-67.)

Abbyy/Lexmark argues that the '342 and '489 Patents compel the construction of the term 'character' to indicate a single, individual symbol or letter because the patents explicitly describe the parsing process to render segments of a line of text into individual characters, which are meant to be both individual and isolated. (*See* '342 Patent at 4:25-27; 15:61-62; '489 Patent at 4:27-29, 16:31-32.) Further, Abbyy/Lexmark contends that in the section of the '342 Patent addressing the instance in which the invention mistakenly grabs the term 'ite' instead of its composite letters, the patent describes the use of templates of single, individual characters to identify single characters by parsing the segment into the characters 'i' and then 't' and following. (*See* '342 Patent at 24:29-49.)

However, in order to provide a consistent construction of the term, the Court would be remiss to limit the construction to a single, individual symbol or letter, where the patent clearly and specifically discloses multi-symbol characters such as "ite" and "ff" and "ffi." (*See* '342 Patent at 24:34-36, 26:59-64; '009 Patent at 23:4-9.)

Accordingly, the Court construes the term "character" to mean: "One or a group of adjacent letters, digits, or other symbols."

**3. "Identifying"**

The word "identifying" appears in Claims 1,4, 6, and 18 of the '342 Patent and also appears in Claim 7 (as dependent on Claim 2) of the '489 Patent.

Nuance asserts that this term does not require construction, while Abbyy/Lexmark asserts that it does. However, Abbyy/Lexmark actually includes the term identifying in its construction of the term itself and tries to read into the term the methods of feature analysis and template matching.

Abbyy/Lexmark argues the term “identifying” should be construed to mean “Determining the identity of an unknown character through feature analysis or template matching or both.” (Statement, Ex. 1.) Nuance asserts that this term is a simple word used in its every day, commonly understood way and should be given its plain and ordinary meaning. *See, e.g.*, *Johnson Worldwide Assocs. v. Zebco Corp.*, 175 F.3d 985, 989 (Fed. Cir. 1999) (“General descriptive terms will ordinarily be given their full meaning; modifiers will not be added to broad terms standing alone.”) In addition, the term identifying appears throughout portions of the patents that do not allude to the specific processing and may be analyzed according to any one of a number of techniques. Therefore, the Court does not find it persuasive to limit the construction of the term to only either feature analysis of template analysis.

Accordingly, the Court construes the term “identifying” to mean: “ identifying.”

**4. “Shape characteristics”**

The word “shape characteristics” appears in Claim 1 of the ’342 Patent. This is the only place this term appears.

Again, Nuance asserts that this term does not require construction, while Abbyy/Lexmark asserts that it does. “Victory in an infringement suit requires a finding that the patent claim covers the alleged infringer’s product or processes, which in turn necessitates a determination of what the claim terms mean.” *Markman*, 517 U.S. at 374. Thus, this Court will construe the term “shape characteristics.”

Abbyy/Lexmark argues the term “shape characteristics” should be construed to mean “Statistical information derived from a horizontal window and a vertical window comprising profile data, polygonal representations, phase change information, and counts of the number of

on pixels in each row of the character." (Statement, Ex. 1.) Nuance again asserts that this term should be given its plain and ordinary meaning without importing numerous limitations from the specifications as attempted by Abbyy/Lexmark in its proposed construction. Abbyy/Lexmark's proposed construction clearly attempts to import the limitations of a shape characteristic analyzing process disclosed in one specific embodiment into the construction of the otherwise generic term. (*See* '342 Patent at 20:45-50.) In that instance, the shape characteristics are extracted from the feature analysis program disclosed in the patent, but it does not allow the Court to import the limitations into the construction of the term. In its argument, Abbyy/Lexmark contends that the shape characteristics are extracted from the feature analysis process in the form of statistical information such as profile data, polygon representations of the characters, phase change information, and counts of the numbers of on pixels in each row of the character. (*See* '342 Patent at 20:47-50.) However, these are limitations of this specific embodiment. "Limitations from the specification (such as from the preferred embodiment) may not be read into the claims, absent the inventor's express intention to the contrary." *Teleflex*, 299 F.3d at 1326 ; *see also CCS Fitness*, 288 F.3d at 1366.

Unlike the holding in *ICU Medical, Inc. v. Alaris Medical Sys., Inc.*, 558 F.3d 1368, 1374 (Fed. Cir. 2009), the Court finds that there is sufficient intrinsic evidence that the term would not include the limitations Abbyy/Lexmark seeks to impose. The patent describes a number of processes and does not restrict the meaning of the term "shape characteristics" to the single description outlined by Abbyy/Lexmarks' proposal. However, the Court finds that the description of the term in its responsive brief is more explicative than the term itself and adopts the construction "features of characters" which are extracted using several different means, as described in the patent.

Accordingly, the Court construes the term "shape characteristics" to mean: "features of characters."

**5. "Feature analysis"**

The phrase "feature analysis" appears in Claims 4, 6, 14, and 18 of the '342 Patent.

United States District Court
For the Northern District of California

Abbyy/Lexmark argues that the term "feature analysis" should be construed to mean "An analysis of statistical information derived from a horizontal window and a vertical window consisting of profile data, polygonal representations, phase change information, and counts of the number of on pixels in each row of the character." (Statement, Ex. 1.) Nuance, on the other hand, argues that the term "feature analysis" should be construed to mean "Recognizing a character in an image by routines that extract features of the character and analyze the features." (*Id.*)

In the specification, in a heading entitled "FEATURE ANALYSIS," the patent sets out that "[t]he preferred embodiment of the present invention discloses a plurality of routines for analyzing the features of images passed as input to the feature analysis process," with each routine called an "isit ... (e.g., 'is it' an a)." ('342 Patent at 19:37-45.) As isit can be implemented in a variety of ways, not just using the polygon fitting and the delineated statistical information in Abbyy/Lexmark's restrictive construction. (*See id*. at 23:18-30.)

Accordingly, the Court construes the term "feature analysis" to mean: "Recognizing a character in an image by routines that extract features of the character and analyze the features."

**6. "Template matching"**

The phrase "template matching" appears in Claims 7, 14, and 18 of the '342 Patent.

Abbyy/Lexmark argues that the term "template matching" should be construed to mean "A character recognition process that compares an unknown character (an image in a window) to each template in a template cache until a match occurs or the cache is exhausted by constructing must-be-on and must-be-off arrays from the unidentified image, making a dimension check, and comparing must-be-on and must-be-off pixel information." (Statement, Ex. 1.) Nuance, on the other hand, argues that the term "template matching" should be construed to mean "A character recognition process in which representations of the patterns, shapes, or images of an unknown character are compared with previously generated representations of the patterns, shapes, or images of known characters." (*Id.*)

Case3:08-cv-02912-JSW Document332 Filed06/15/11 Page12 of 15



Here, again, Abbyy/Lexmark tries to import a limitation from the specification for the term template to necessarily include the 'must-be-on' or 'must-be-off' feature. This construction has already been rejected by the Court. Secondly, Abbyy/Lexmark seeks to have the "until a match occurs or the cache is exhausted" become part of the construction of the term template matching. The Court does not find an explicit or express intent to add in a limitation that the template comparison stops once a match is found. Rather, the patent appears to indicate that the matching continues to occur within the library cache regardless of whether a single match is found. There is no indication in the intrinsic evidence that the inventors expressed a clear intent to restrict or exclude template matching only until a match is found. *See Liebel-Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 909 (Fed. Cir. 2004) ("Absent a clear disclaimer of particular subject matter, the fact that the inventor may have anticipated that the invention would be used in a particular way does not mean that the scope of the invention is limited to that context." ) (internal quotation omitted). The Court declines to limit the claim to the preferred embodiment disclosed in the specification.

Accordingly, the Court construes the term "template analysis" to mean: "A character recognition process in which representations of the patterns, shapes, or images of an unknown character are compared with previously generated representations of the patterns, shapes, or images of known characters."

**7. "Second character recognition process"**

The phrase "second character recognition process" appears in Claim 7 of the '342 Patent.

Abbyy/Lexmark argues that the term "second character recognition process" should be construed to mean "feature analysis process." (Statement, Ex. 1.) Nuance, on the other hand, argues that the term "second character recognition process" should be construed to mean "A process for recognizing a character that is different than the first character recognition process." (*Id.*)

The key difference in the parties' proposals is that Abbyy/Lexmark attempts to limit the construction of the term to be only feature analysis and not any other possible recognition process. However, this construction would render meaningless the independent claims 7 and 13 (as opposed to claim 18). Accordingly, under the theory of claim differentiation, the Court declines to adopt Abbyy/Lexmark's restrictive construction. *See Andersen*, 474 F.3d at 1369.

Accordingly, the Court construes the term "second character recognition process" to mean: "A process for recognizing a character that is different than the first character recognition process."

**8. "Portion of said image"**

The phrase "portion of said image" appears in Claim 18 of the '342 Patent.

Abbyy/Lexmark argues that the term "portion of said image" should be construed to mean "A representation of a character." (Statement, Ex. 1.) Nuance asserts that the term is simple and additional construction would amount to meaningless verbiage. The figures of the patent set illustrate a "portion of a page" and "the same portion of a page." ('342 Patent at Figs. 3(a) and 3(b).) (*See* '342 Patent at 6:10-18.) The figures include parts of a letter and parts of a drawing. In addition, Claim 18 provides that "a portion of said image, said portion *including* a representation of said character." (*Id.* at 30:28-31.) With the word 'including' in the same phrase of the claim, it appears that the definition of a portion of said image would include more than simply the representation of a character.

Accordingly, the Court construes the term "a portion of said image" to mean: "a portion of said image."

**9. "Determining a value related to the probability of occurrence"**

The phrase "determining a value related to the probability of occurrence" appears in Claim 22 (as dependent on Claims 1 and 21) of the '009 Patent.

Abbyy/Lexmark argues that the term "determining a value related to the probability of occurrence" should be construed to mean "Computation through modeling language as a second order Markov process." (Statement, Ex. 1.) Nuance, on the other hand, argues that the term

"determining a value related to the probability of occurrence" should be construed to mean "Computing a value related to the chance of occurrence within a reference sequence." (*Id.*) Abbyy/Lexmark, during oral argument before the Court, contended that the construction should allude to the purpose of the patent and should reflect that the computation is dependent on probabilities. However, its proposed construction improperly imports limitations from the specifications and there is no principal of construction which requires that the Court delineate the alleged purpose of the patent within its construction of any particular term.

Accordingly, the Court construes the term "Determining a value related to the probability of occurrence" to mean: "Computing a value related to the chance of occurrence within a reference sequence."

**10. "Input window"**

The phrase "input window" also appears in Claim 22 (as dependent on Claims 1 and 21) of the '009 Patent.

Abbyy/Lexmark argues that the term "input window" should be construed to mean "A fixed-length portion of a string of characters that is shifted to the right by one character from the beginning of the string to the end of the string." (Statement, Ex. 1.) Nuance, on the other hand, argues that the term "input window" should be construed to mean "A plurality of adjacent character candidates being processed." (*Id.*)

Again, Abbyy/Lexmark's proposed construction improperly seeks to import limitations from the specifications. The patent does not require shifting and specifically permits the window to be any length. (*See* '009 Patent at 23:12-18.)

Accordingly, the Court construes the term "input window" to mean: "A plurality of adjacent character candidates being processed."

**11. "Groups of Elements"**

The phrase "groups of elements" also appears in Claim 22 (as dependent on Claims 1 and 21) of the '009 Patent.

Abbyy/Lexmark argues that the term "groups of elements" should be construed to mean "Groups of characters that facilitate the generation of statistics for computing ngram probabilities." (Statement, Ex. 1.) Nuance asserts that this term should be construed to mean "groups of characters." (*Id.*)

Again, Abbyy/Lexmark, without reasoning, attempts to have the Court adopt the limitations from the specifications. During oral argument, Abbyy/Lexmark agreed to Nuance's proposed construction but requested, without legal authority, that the Court also incorporate the probability statistics contained in the table of column seven of the '009 Patent. There is no reasoned basis upon which the Court can incorporate into the construction of a claim term an illustrative example set out in the specifications. Nuance, on the other hand, proposes a clear construction that reflects the scope of the patent. (*See, e.g.*, '009 Patent at 6:6-9.)

Accordingly, the Court construes the term "groups of elements" to mean: "groups of characters."

**CONCLUSION**

Based on the analysis set forth above, the Court adopts the foregoing constructions of the disputed terms. The parties are ordered to submit a further joint case management report pursuant to Patent Standing Order ¶ 13 by no later than July 6, 2011.

**IT IS SO ORDERED.**

Dated: June 15, 2011

JEFFREY S. WHITE
UNITED STATES DISTRICT JUDGE

Case: 14-1629 Document: 20 Page: 74 Filed: 11/14/2014

IN THE UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA

NUANCE COMMUNICATIONS INC,

Plaintiff,

v.

ABBYY SOFTWARE HOUSE, et al.,

Defendants.

______________________________________/

No. C 08-02912 JSW

**ORDER RESOLVING PRETRIAL SUBMISSIONS**

Now before the Court are several issues related to the parties' pretrial submissions. The Court HEREBY ORDERS as follows:

(1) Nuance Communications, Inc. may call Sergy Andreev and Dean Tang during the principal presentation of its case-in-chief;

(2) the Court shall not resolve the parties' multiple discovery designation disputes, deposition designations, or objections to exhibits prior to introduction of such evidence at trial (and any time utilized to resolve any such objection shall be deducted from the losing party's overall trial time);

(3) the Court adopts the plain and ordinary meaning of the terms "identifying" and "recognizing" as the same: "to establish the identity of."

**IT IS SO ORDERED.**

Dated: August 6, 2013

Jeffrey S White

JEFFREY S. WHITE
UNITED STATES DISTRICT JUDGE

United States District Court
For the Northern District of California

IN THE UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA

NUANCE COMMUNICATIONS, INC.,

Plaintiff,

v.

ABBYY SOFTWARE HOUSE, ET AL.,

Defendants.

_______________________________________/

No. C 08-02912 JSW

**FINAL JUDGMENT**

Pursuant to the jury's verdict, entered on August 26, 2013, JUDGMENT is HEREBY ENTERED in favor of Defendants and against Plaintiff.

**IT IS SO ORDERED.**

Dated: August 26, 2013

JEFFREY S. WHITE
UNITED STATES DISTRICT JUDGE

United States District Court
For the Northern District of California

IN THE UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA

NUANCE COMMUNICATIONS, INC.,

Plaintiff,

v.

ABBYY SOFTWARE HOUSE, ET AL.,

Defendants.

_______________________________________/

No. C 08-02912 JSW

**ORDER GRANTING MOTION FOR COSTS**

Now before the Court is the motion by Defendants ABBYY USA Software House, Inc., ABBYY Software, Ltd., ABBYY Production, LLC, and Lexmark International, Inc. ("Defendants") to compel payment of costs award and request entry of final judgment. This motion is fully briefed and ripe for decision, and the Court finds the motion suitable for disposition without oral argument. N.D. Civ. L.R. 7-1(b). Accordingly, the hearing set for June 6, 2014, is VACATED. Having reviewed the parties' pleadings, the relevant legal authority, and the record in this case, for the reasons set forth herein, the Court GRANTS the motion for payment of costs award.

After a full and fair trial on the issues selected by Nuance for its case-in-chief, the Court entered judgment on August 26, 2013. The final judgment explicitly stated that the judgment was "entered in favor of Defendants and against Plaintiff." The judgment did not exempt any of Nuance's causes of action or reserve judgment on any of Nuance's patents that it chose not to pursue at trial. Costs are due to the prevailing party.

Case3:08-cv-02912-JSW Document1003 Filed06/04/14 Page4 of 3

During the course of the litigation, the Court afforded Nuance the opportunity to pursue discovery and claim construction on all its patents. However, for the purposes of case management, the Court agreed with Nuance's proposal that it would conduct a single trial to allow the plaintiff to pursue a "manageable set" of patents and claims for "one trial." (Transcript of Proceedings, July 2, 2010 (docket 218) at 15-16.) The Court had the parties employ the services of a special master to winnow the claims to a manageable set and to aid the parties in configuring the issues presented during the single trial. Although in the initial stages of this case, the Court kept the option open to Nuance to pursue discovery and claims construction on all of its originally asserted patents, there was never any mention that there would be serial trials. Rather, according to its own representations, Nuance selected its "best" and "strongest" patents for trial. (Transcript of Proceedings, April 9, 2010 at 10.)

In a complex case such as this, the Court, with the aid and cooperation of the parties, may administer the proceeding in the best way possible to bring the case to trial before a jury. *See e.g., In re Katz Interactive Call Processing Patent Litigation*, 639 F.3d 1303, 1313 (9th Cir. 2011) (holding that, given the district court's broad discretion to manage the cases before it and the "strong public interest in the finality of judgments in patent litigation," the Ninth Circuit declined to adopt appellant's position that it should be able to seek further disposition on its previously unselected claims). Here, Nuance cooperatively selected its best and strongest patents for trial after full discovery and unrestricted claims construction selection process.

At the end of a lengthy trial, the jury's verdict was solely in Defendants' favor. Accordingly, final judgment in Defendants' favor was entered. After the verdict and entry of judgment, Defendants' filed their bills of cost. Nuance filed a renewed motion for judgment as a matter of law, or in the alternative, for a new trial. Not once did Nuance indicate that the trial was not complete. Nuance did not indicate any intention to try its remaining, unasserted patents or claims. The Court denied the motion for judgment as a matter of law. Once undisputed costs were assessed, the Court referred Defendants' motion for review of the taxation of disputed costs to a magistrate for a report and recommendation. On January 9, 2014, Magistrate Judge Nandor Vadas issued his report and recommendation recommending the award of $195,239.51

Case3:08-cv-02912-JSW Document1003 Filed06/04/14 Page3 of 3



in contested costs. Defendants sought clarification on the additional, undisputed costs, which was granted on February 3, 2014, for a total sum of reimburseable costs in the amount of $298,389.68. Nuance did not file objections to the Magistrate's reports.

Considering the parties' cooperative and voluntary management of the case for trial, the Court's entry of final judgment, Nuance's failure to mention in its post-trial motions or its notice of appeal its potential request for a new trial on additional patents and claims, and Nuance's failure to file any timely objections to the Magistrate's reports, the Court finds entry of judgment appropriate on all of Nuance's patents and orders Nuance to pay forthwith the costs as recommended by Magistrate Judge Vadas in the amount of $298,389.68. The Court does not award Defendants the cost of bringing this motion.

**IT IS SO ORDERED.**

Dated: June 4, 2014

JEFFREY S. WHITE
UNITED STATES DISTRICT JUDGE

US006038342A


# United States Patent [19]

**Bernzott et al.**

[11] **Patent Number: 6,038,342**

[45] **Date of Patent: Mar. 14, 2000**


[54] **OPTICAL CHARACTER RECOGNITION METHOD AND APPARATUS**


[75] Inventors: **Phillip Bernzott**; **John Dilworth**; **David George**, all of Oakland; **Bryan Higgins**; **Jeremy Knight**, both of Berkeley, all of Calif.

[73] Assignee: **Caere Corporation**, Los Gatos, Calif.



[21] Appl. No.: **08/114,608**

[22] Filed: **Aug. 31, 1993**

**Related U.S. Application Data**

[60] Continuation of application No. 07/914,120, Jul. 15, 1992, abandoned, which is a division of application No. 07/799, 549, Nov. 27, 1991, Pat. No. 5,278,918, which is a continuation of application No. 07/230,847, Aug. 10, 1988, Pat. No. 5,131,053.

[51] **Int. Cl.**[7] ........ **G06K 9/34**

[52] **U.S. Cl.** ........ **382/173**

[58] **Field of Search** ........ 382/14, 15, 25, 382/30, 34, 33



[56] **References Cited**

U.S. PATENT DOCUMENTS

| | | | |
|---|---|---|---|
| 3,104,369 | 9/1963 | Rabinow et al. | 382/30 |
| 3,713,099 | 1/1973 | Henstreet | 340/149 |
| 3,713,100 | 1/1973 | Henstreet | 340/149 |
| 3,717,848 | 2/1973 | Irvin et al. | 382/30 |
| 3,873,972 | 3/1975 | Levine | 382/14 |
| 3,930,231 | 12/1975 | Henrichan, Jr. et al. | 382/30 |
| 4,027,284 | 5/1977 | Hoshino et al. | 382/30 |
| 4,499,596 | 2/1985 | Casey et al. | 382/15 |
| 4,672,678 | 6/1987 | Koezuka et al. | 382/30 |
| 4,850,026 | 7/1989 | Jeng et al. | 380/14 |
| 4,887,304 | 12/1989 | Terzian | 382/30 |
| 4,944,022 | 7/1990 | Yasujima et al. | 382/14 |
| 5,014,327 | 5/1991 | Potter et al. | 382/14 |



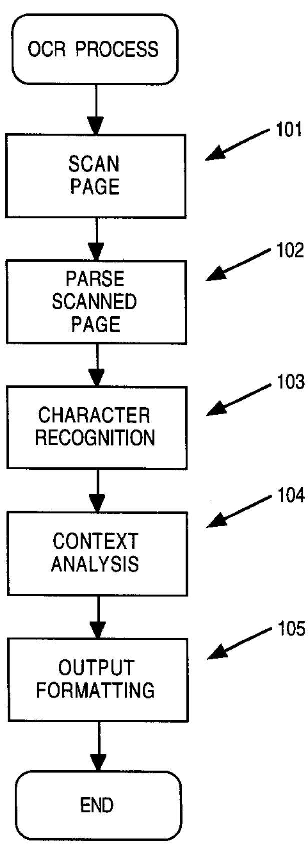
*Primary Examiner*—Jose L. Couso
*Attorney, Agent, or Firm*—Blakely, Sokoloff, Taylor & Zafman LLP



[57] **ABSTRACT**

A system for recognition of characters on a medium. The system includes a scanner for scanning a medium such as a page of printed text and graphics and producing a bit-mapped representation of the page. The bit-mapped representation of the page is then stored in a memory means such as the memory of a computer system. A processor processes the bit-mapped image to produce an output comprising coded character representations of the text on the page. The present invention discloses parsing a page to allow for production of the output characters in a logical sequence, a combination of feature detection methods and template matching methods for recognition of characters and a number of methods for feature detection such as use of statistical data and polygon fitting.


**18 Claims, 21 Drawing Sheets**



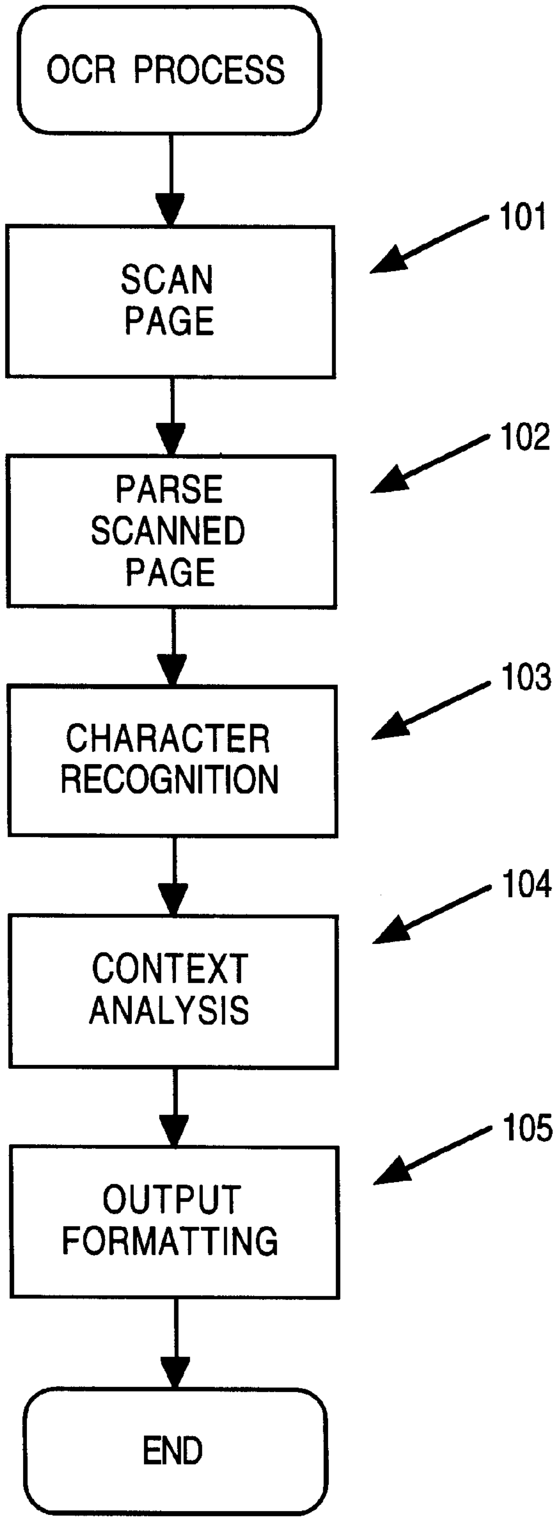

OCR PROCESS
SCAN PAGE
101
PARSE SCANNED PAGE
102
CHARACTER RECOGNITION
103
CONTEXT ANALYSIS
104
OUTPUT FORMATTING
105
END

FIG. 1

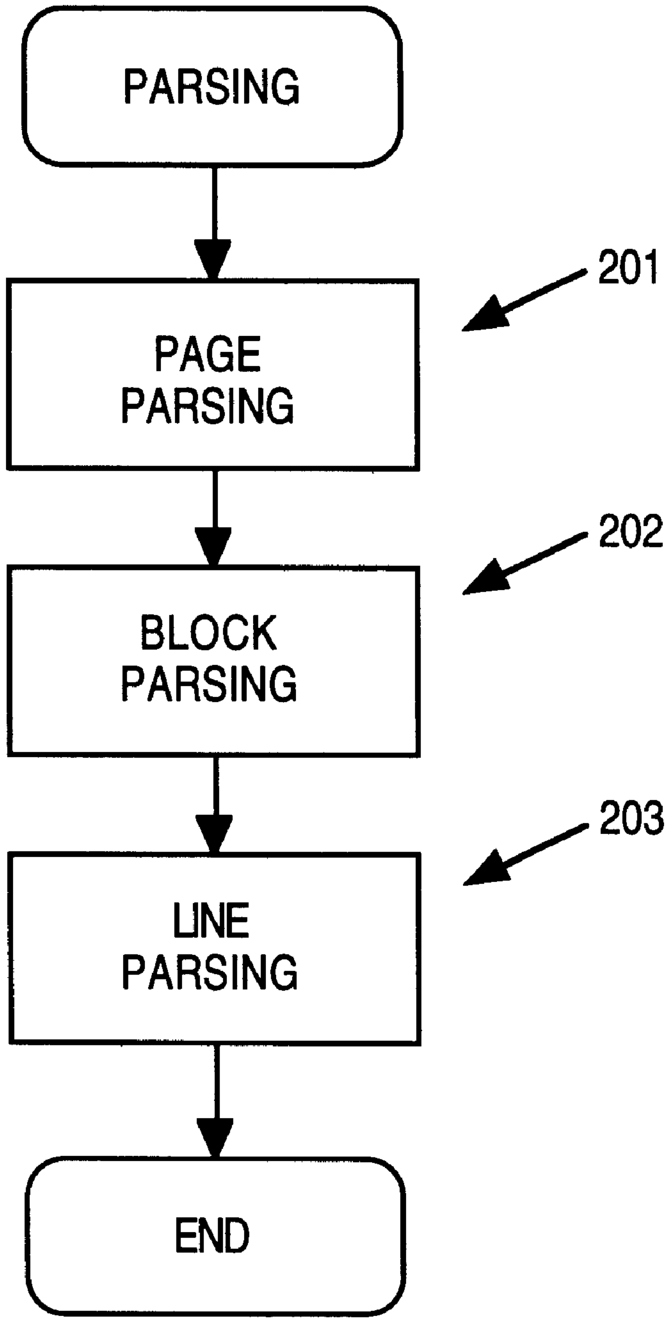
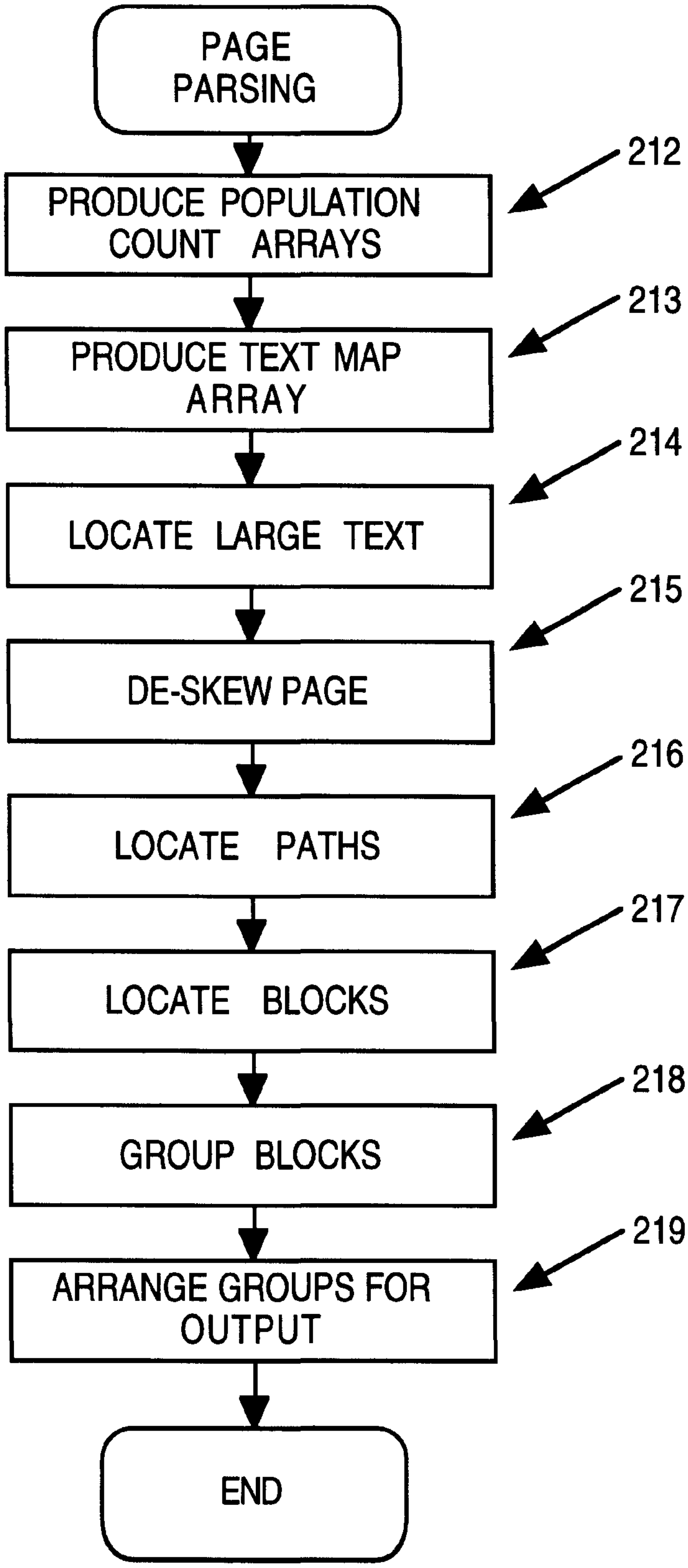

PARSING
PAGE PARSING
201
BLOCK PARSING
202
LINE PARSING
203
END

FIG. 2A



PAGE PARSING
PRODUCE POPULATION COUNT ARRAYS
212
PRODUCE TEXT MAP ARRAY
213
LOCATE LARGE TEXT
214
DE-SKEW PAGE
215
LOCATE PATHS
216
LOCATE BLOCKS
217
GROUP BLOCKS
218
ARRANGE GROUPS FOR OUTPUT
219
END


FIG. 2B

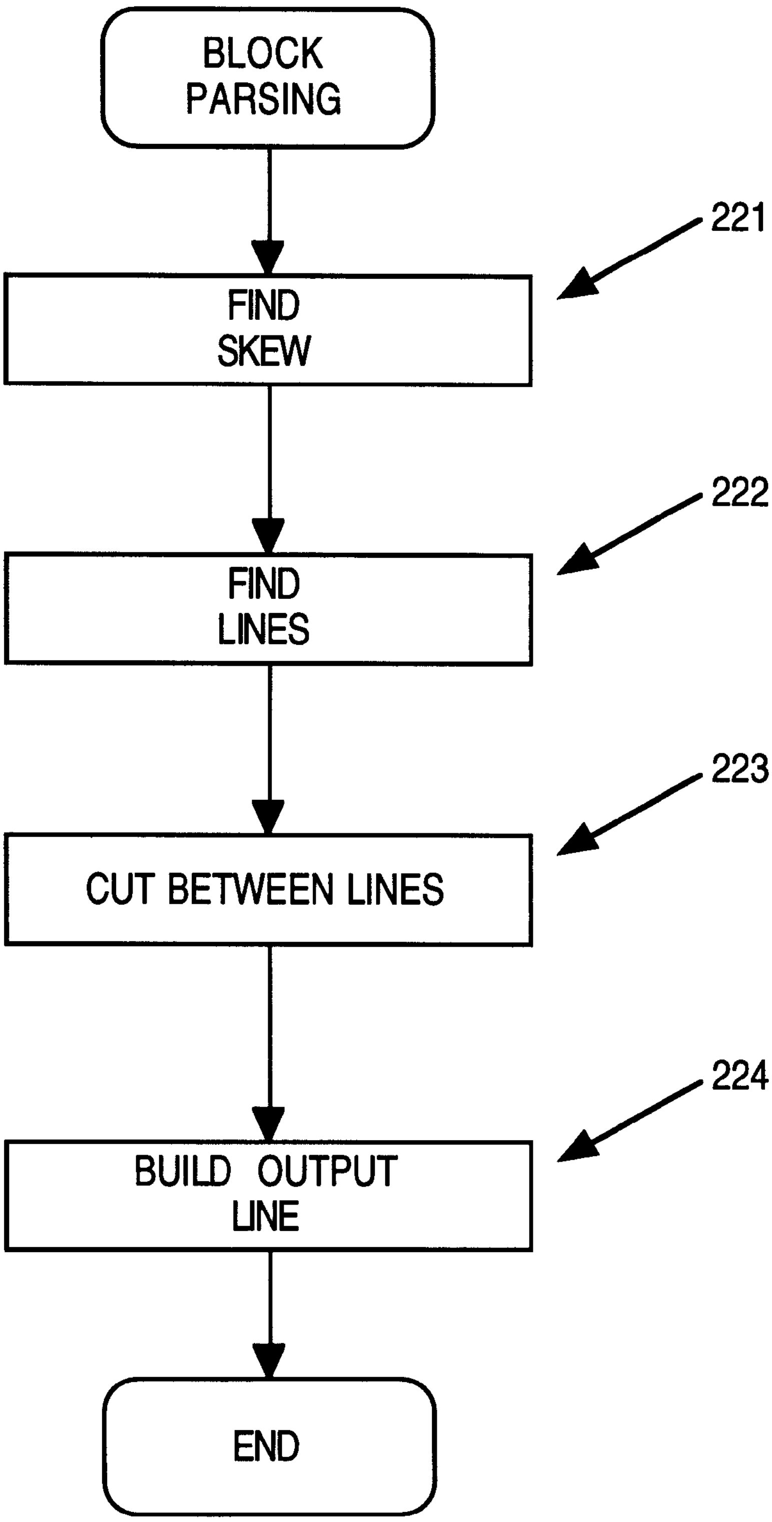

BLOCK
PARSING
221
FIND
SKEW
222
FIND
LINES
223
CUT BETWEEN LINES
224
BUILD OUTPUT
LINE
END

FIG. 2C

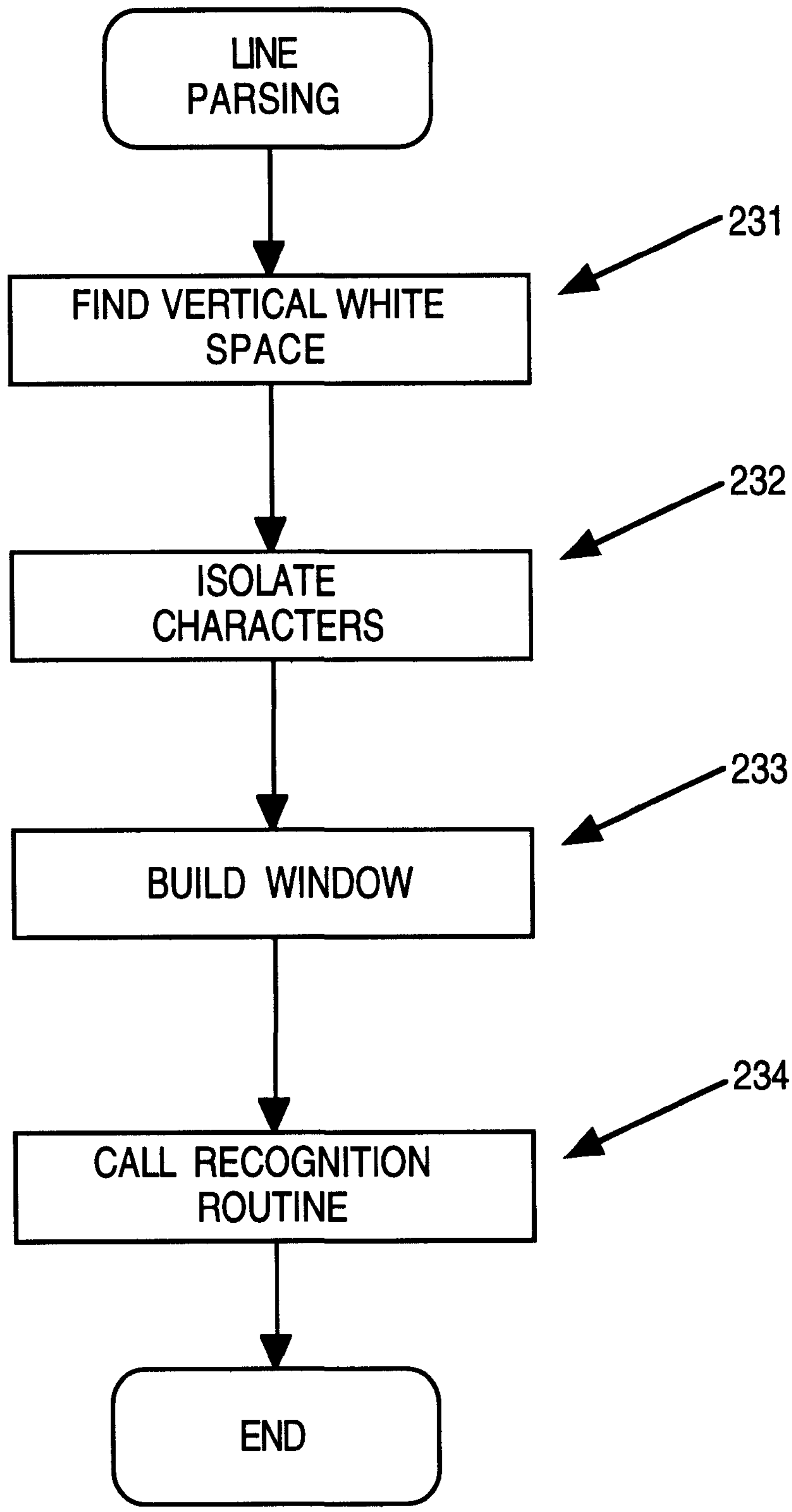

LINE PARSING
231
FIND VERTICAL WHITE SPACE
232
ISOLATE CHARACTERS
233
BUILD WINDOW
234
CALL RECOGNITION ROUTINE
END

FIG. 2D

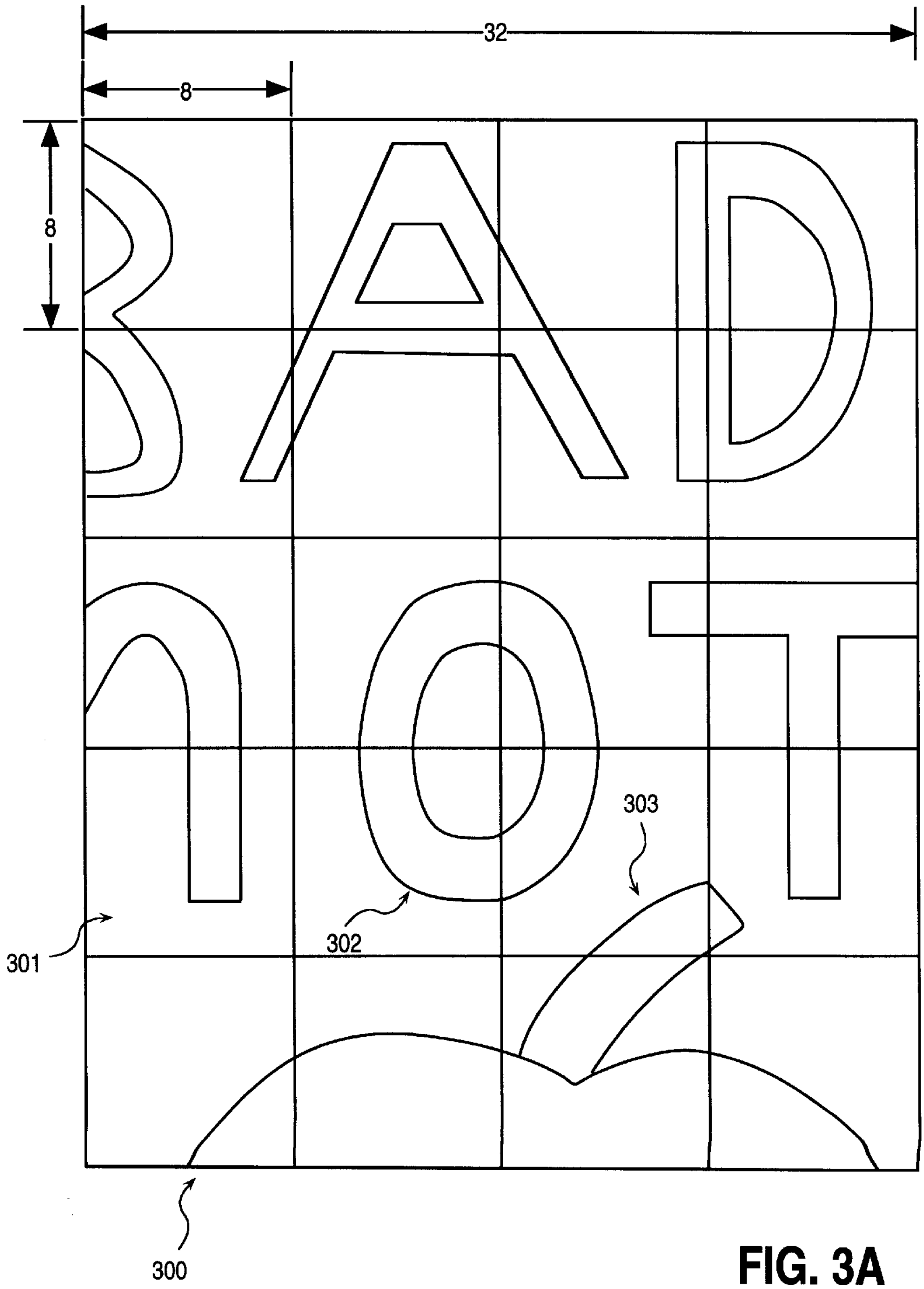
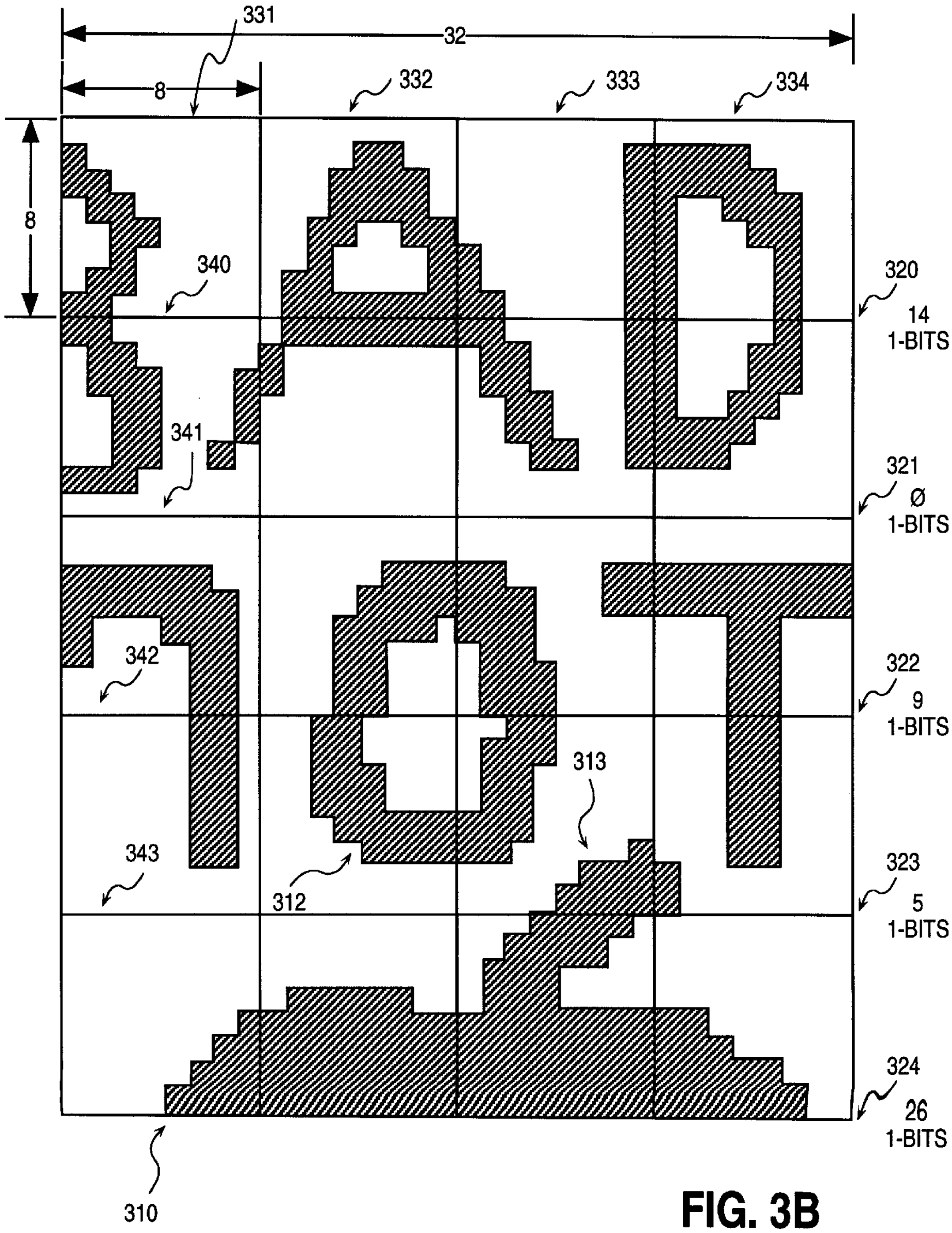



FIG. 3A



331
32
8
332
333
334
8
340
320
14
1-BITS
341
321
Ø
1-BITS
342
322
9
1-BITS
313
343
312
323
5
1-BITS
324
26
1-BITS
310


FIG. 3B

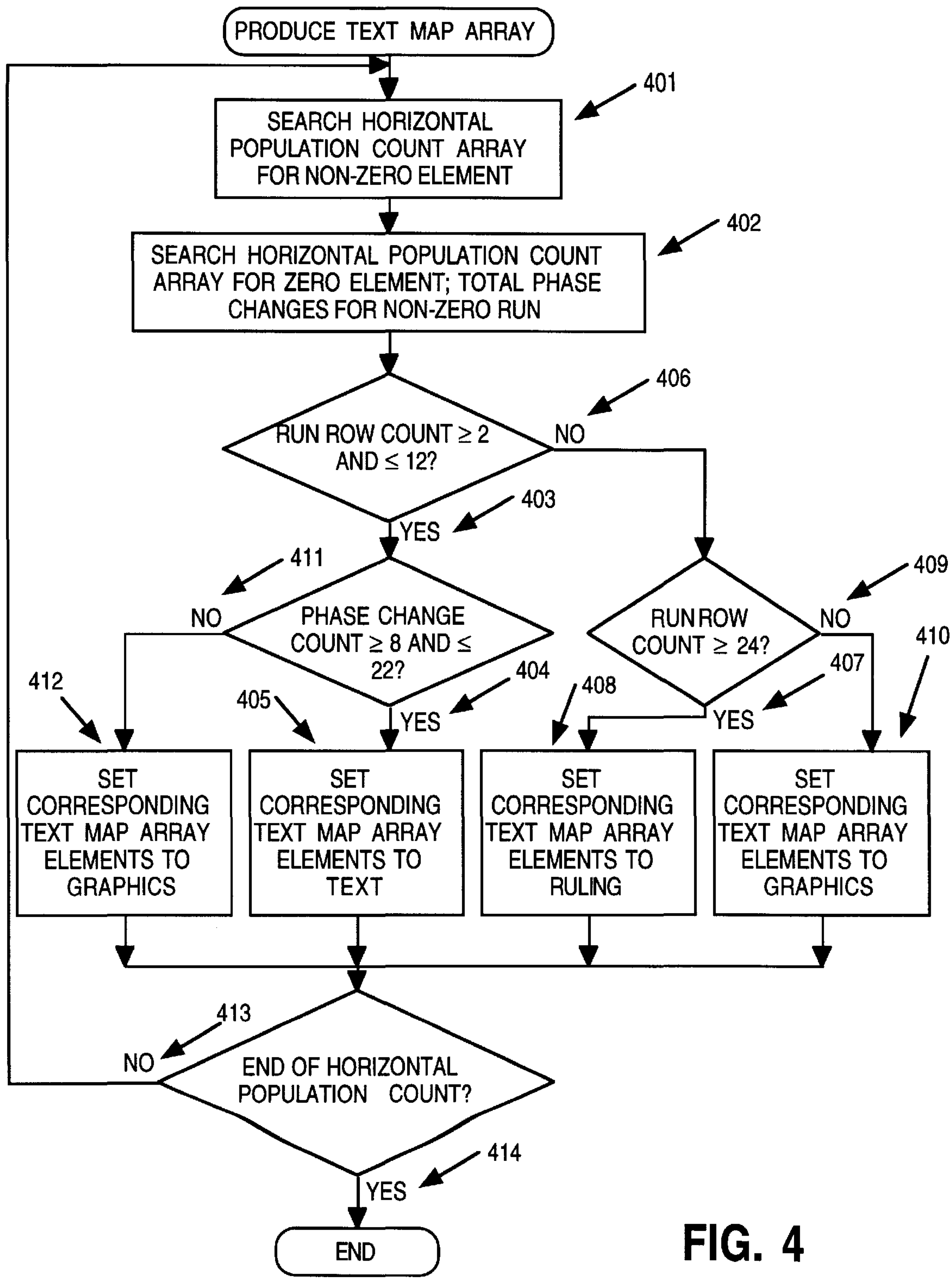

PRODUCE TEXT MAP ARRAY
401
SEARCH HORIZONTAL POPULATION COUNT ARRAY FOR NON-ZERO ELEMENT
402
SEARCH HORIZONTAL POPULATION COUNT ARRAY FOR ZERO ELEMENT; TOTAL PHASE CHANGES FOR NON-ZERO RUN
406
RUN ROW COUNT ≥ 2 AND ≤ 12?
NO
403
YES
411
NO
PHASE CHANGE COUNT ≥ 8 AND ≤ 22?
409
RUN ROW COUNT ≥ 24?
NO
410
412
405
404
YES
408
407
YES
SET CORRESPONDING TEXT MAP ARRAY ELEMENTS TO GRAPHICS
SET CORRESPONDING TEXT MAP ARRAY ELEMENTS TO TEXT
SET CORRESPONDING TEXT MAP ARRAY ELEMENTS TO RULING
SET CORRESPONDING TEXT MAP ARRAY ELEMENTS TO GRAPHICS
413
NO
END OF HORIZONTAL POPULATION COUNT?
414
YES
END

FIG. 4

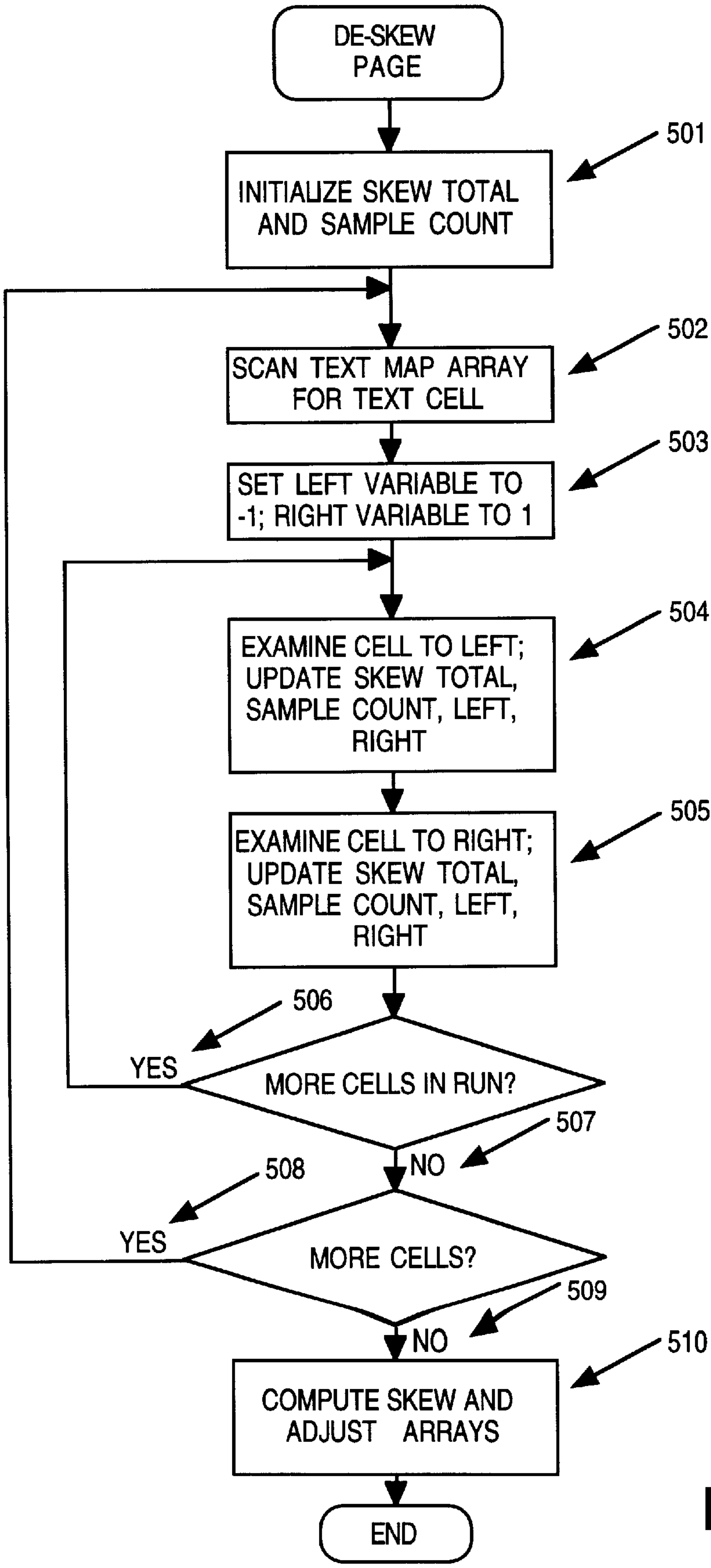
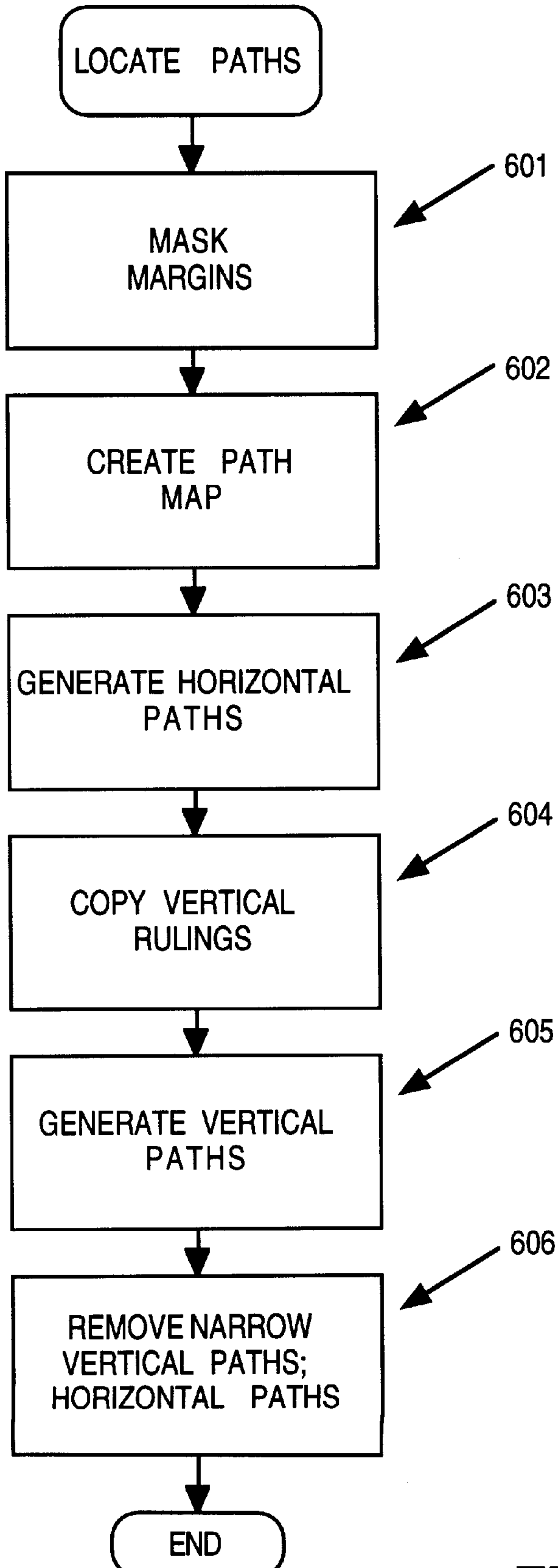
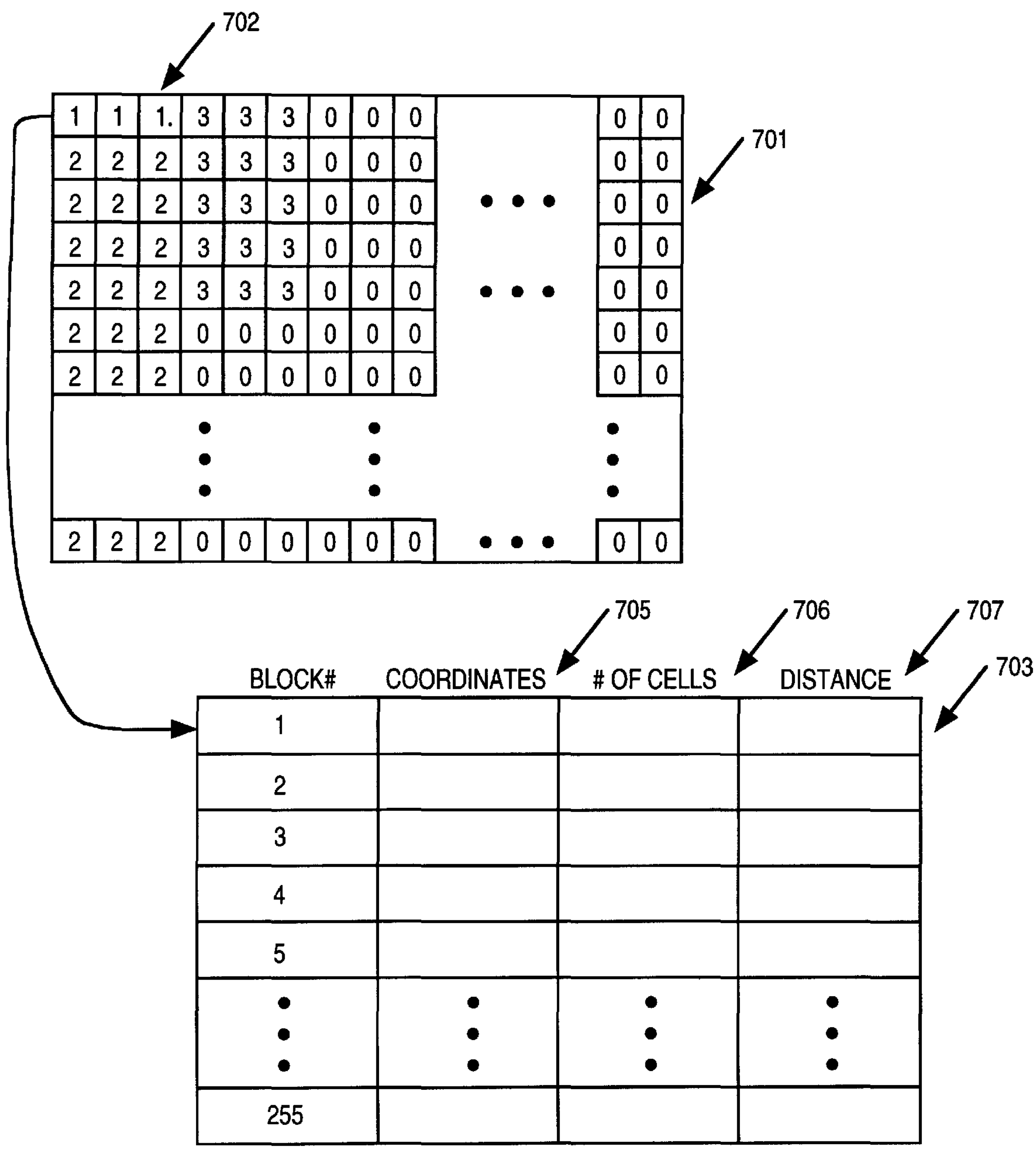

DE-SKEW PAGE
INITIALIZE SKEW TOTAL AND SAMPLE COUNT
501
SCAN TEXT MAP ARRAY FOR TEXT CELL
502
SET LEFT VARIABLE TO -1; RIGHT VARIABLE TO 1
503
EXAMINE CELL TO LEFT; UPDATE SKEW TOTAL, SAMPLE COUNT, LEFT, RIGHT
504
EXAMINE CELL TO RIGHT; UPDATE SKEW TOTAL, SAMPLE COUNT, LEFT, RIGHT
505
506
YES
MORE CELLS IN RUN?
507
NO
508
YES
MORE CELLS?
509
NO
COMPUTE SKEW AND ADJUST ARRAYS
510
END

FIG. 5



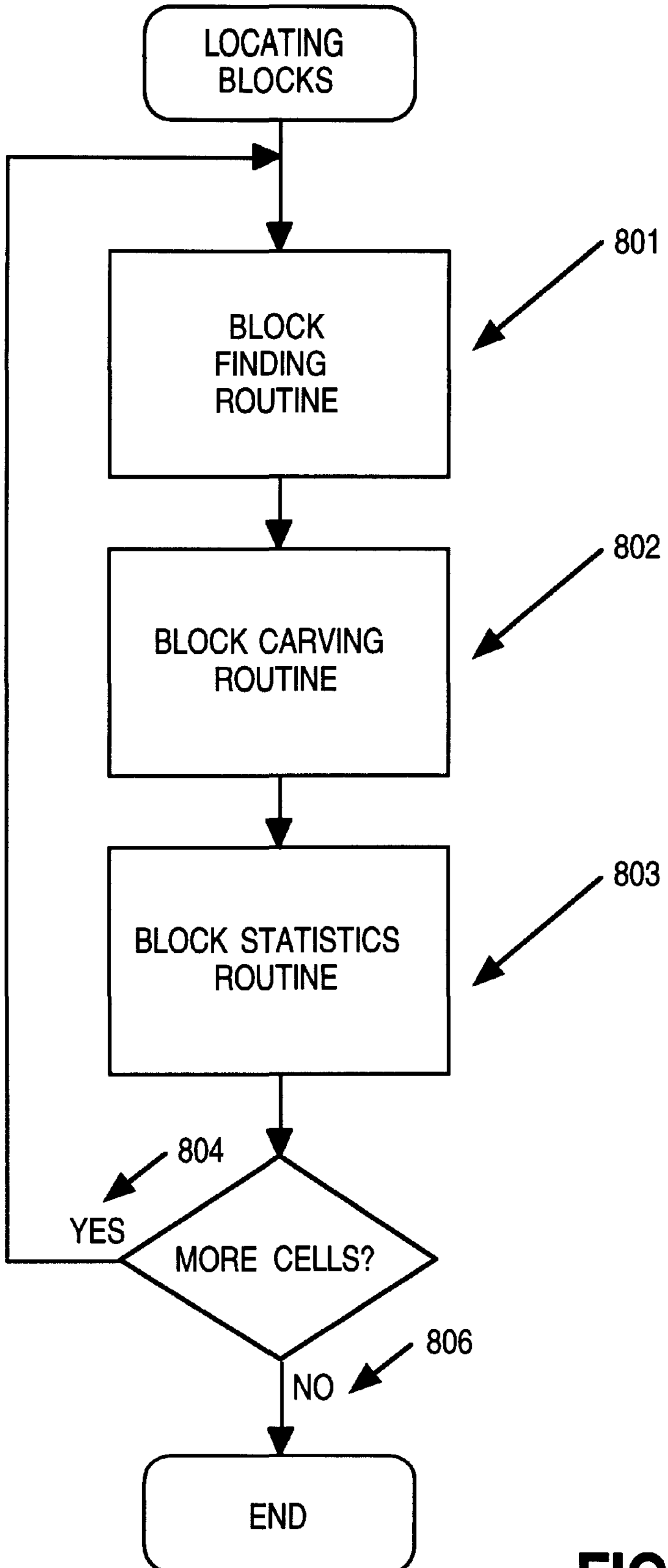
LOCATE PATHS
601
MASK MARGINS
602
CREATE PATH MAP
603
GENERATE HORIZONTAL PATHS
604
COPY VERTICAL RULINGS
605
GENERATE VERTICAL PATHS
606
REMOVE NARROW VERTICAL PATHS; HORIZONTAL PATHS
END


FIG. 6



702
701
1 1 1. 3 3 3 0 0 0 0 0
2 2 2 3 3 3 0 0 0 0 0
2 2 2 3 3 3 0 0 0 0 0
2 2 2 3 3 3 0 0 0 0 0
2 2 2 3 3 3 0 0 0 0 0
2 2 2 0 0 0 0 0 0 0 0
2 2 2 0 0 0 0 0 0 0 0
2 2 2 0 0 0 0 0 0 0 0
705
706
707
703
BLOCK#
COORDINATES
# OF CELLS
DISTANCE
1
2
3
4
5
255


FIG. 7



LOCATING BLOCKS
801
BLOCK FINDING ROUTINE
802
BLOCK CARVING ROUTINE
803
BLOCK STATISTICS ROUTINE
804
YES
MORE CELLS?
806
NO
END


FIG. 8

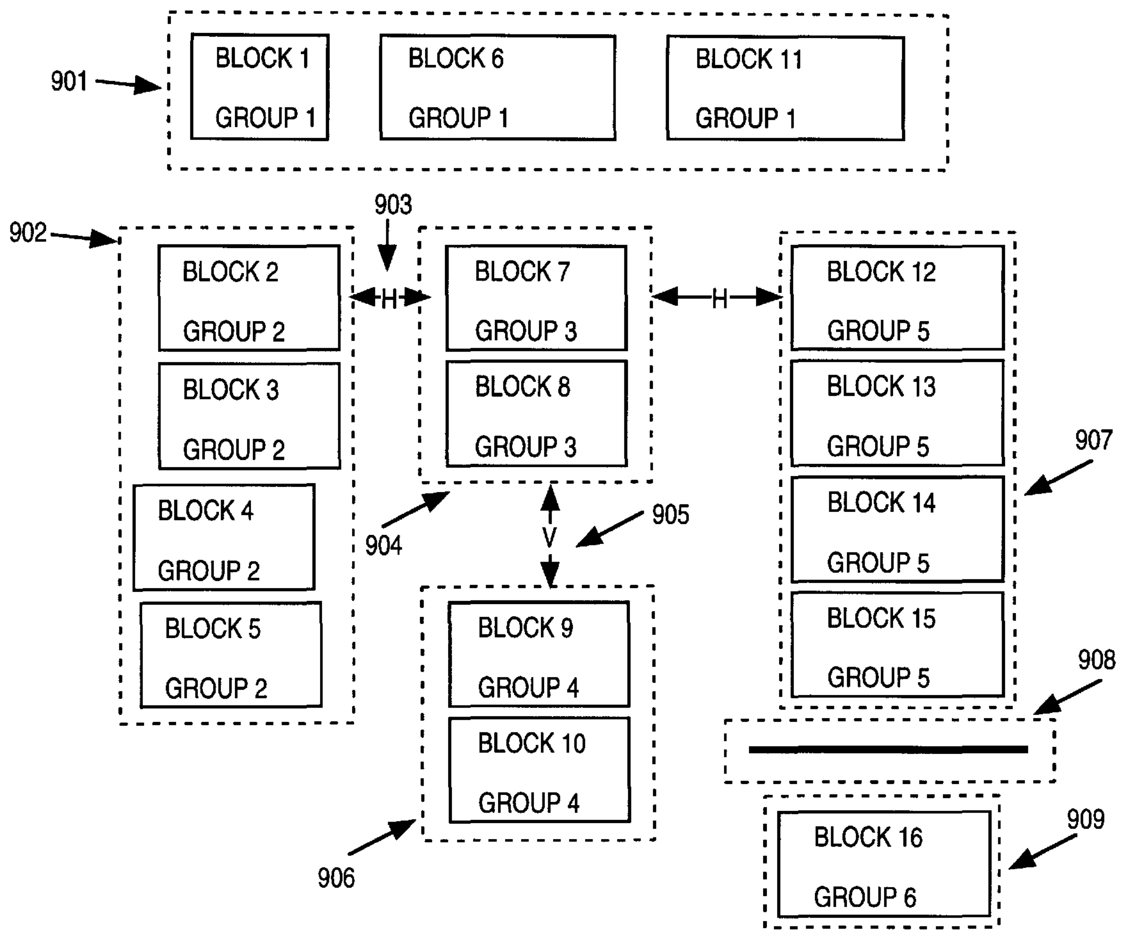

901
BLOCK 1
GROUP 1
BLOCK 6
GROUP 1
BLOCK 11
GROUP 1
902
903
BLOCK 2
GROUP 2
H
BLOCK 7
GROUP 3
H
BLOCK 12
GROUP 5
BLOCK 3
GROUP 2
BLOCK 8
GROUP 3
BLOCK 13
GROUP 5
907
BLOCK 4
GROUP 2
904
V
905
BLOCK 14
GROUP 5
BLOCK 5
GROUP 2
BLOCK 9
GROUP 4
BLOCK 15
GROUP 5
908
BLOCK 10
GROUP 4
BLOCK 16
GROUP 6
909
906

FIG. 9

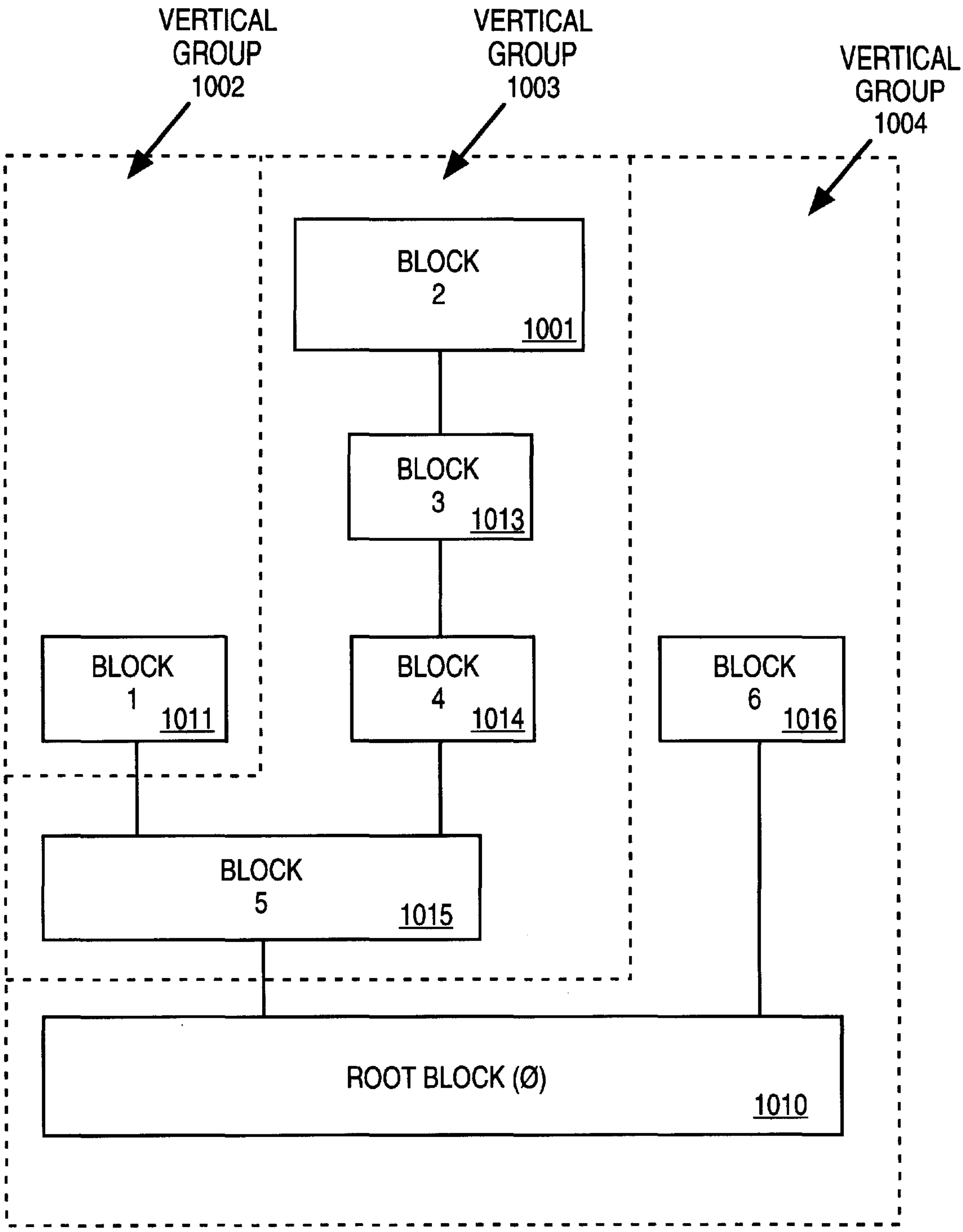

VERTICAL GROUP 1002
VERTICAL GROUP 1003
VERTICAL GROUP 1004
BLOCK 2 1001
BLOCK 3 1013
BLOCK 1 1011
BLOCK 4 1014
BLOCK 6 1016
BLOCK 5 1015
ROOT BLOCK (Ø) 1010

FIG. 10A

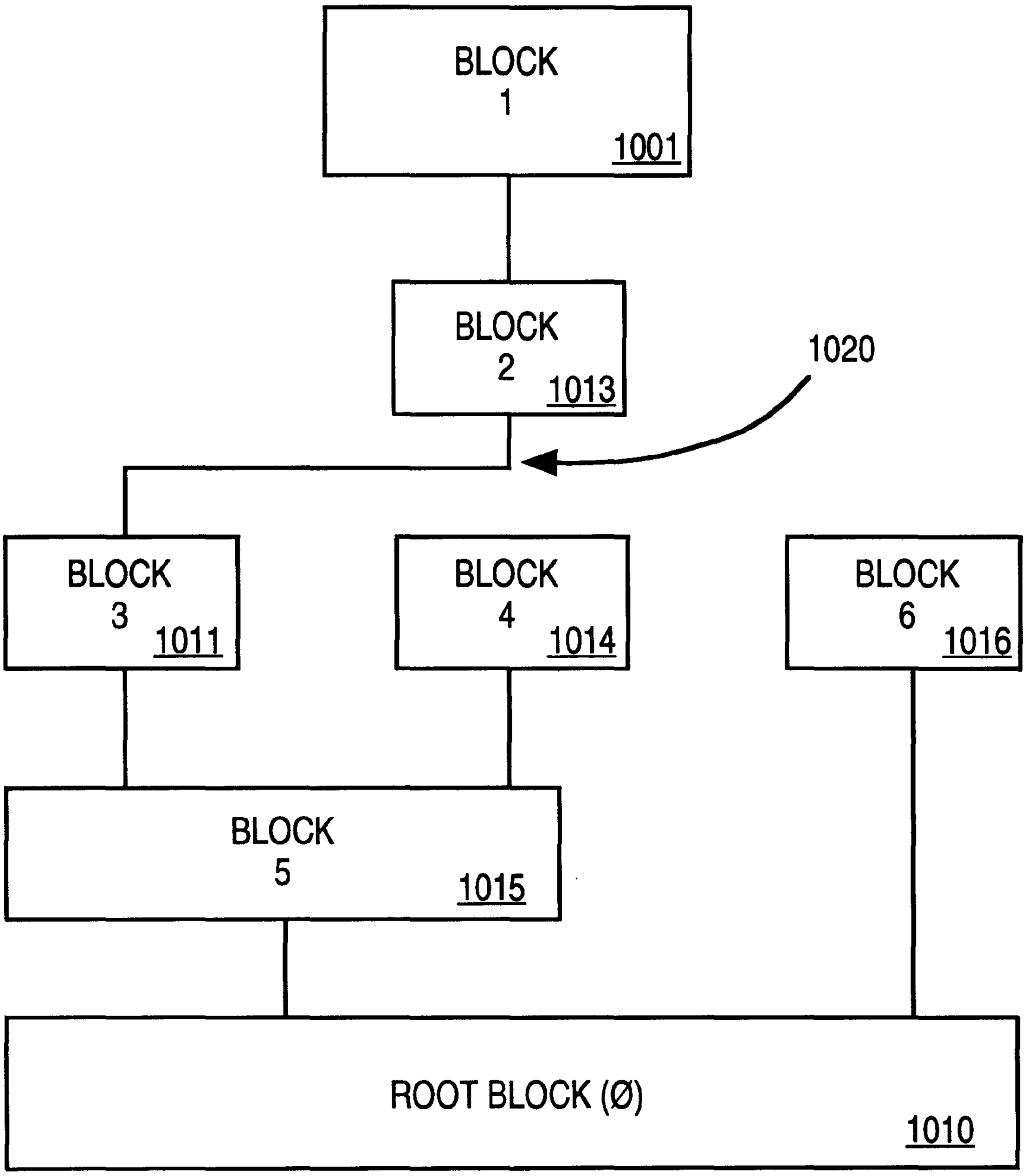

BLOCK
1
1001
BLOCK
2
1013
1020
BLOCK
3
1011
BLOCK
4
1014
BLOCK
6
1016
BLOCK
5
1015
ROOT BLOCK (Ø)
1010

FIG. 10B

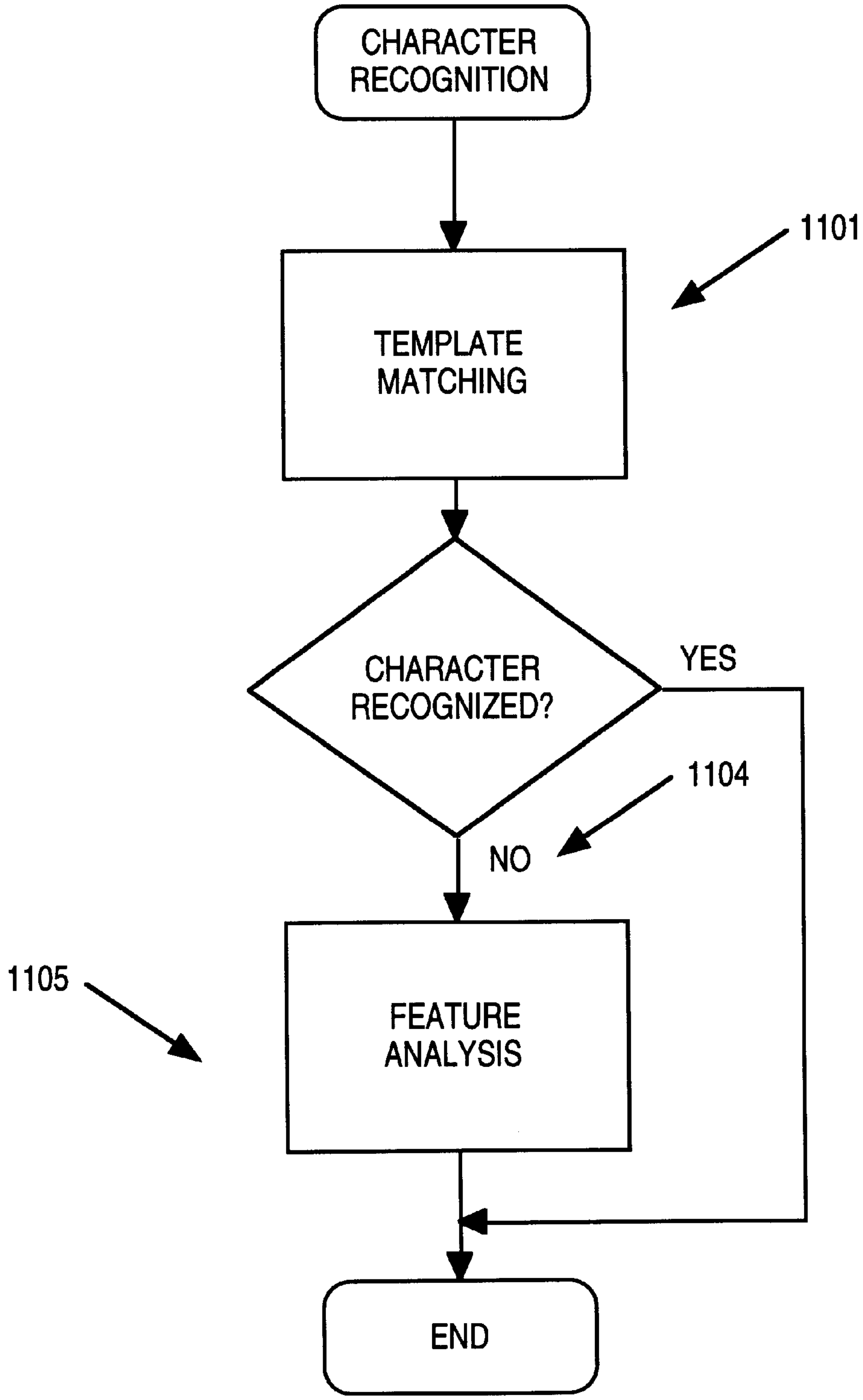
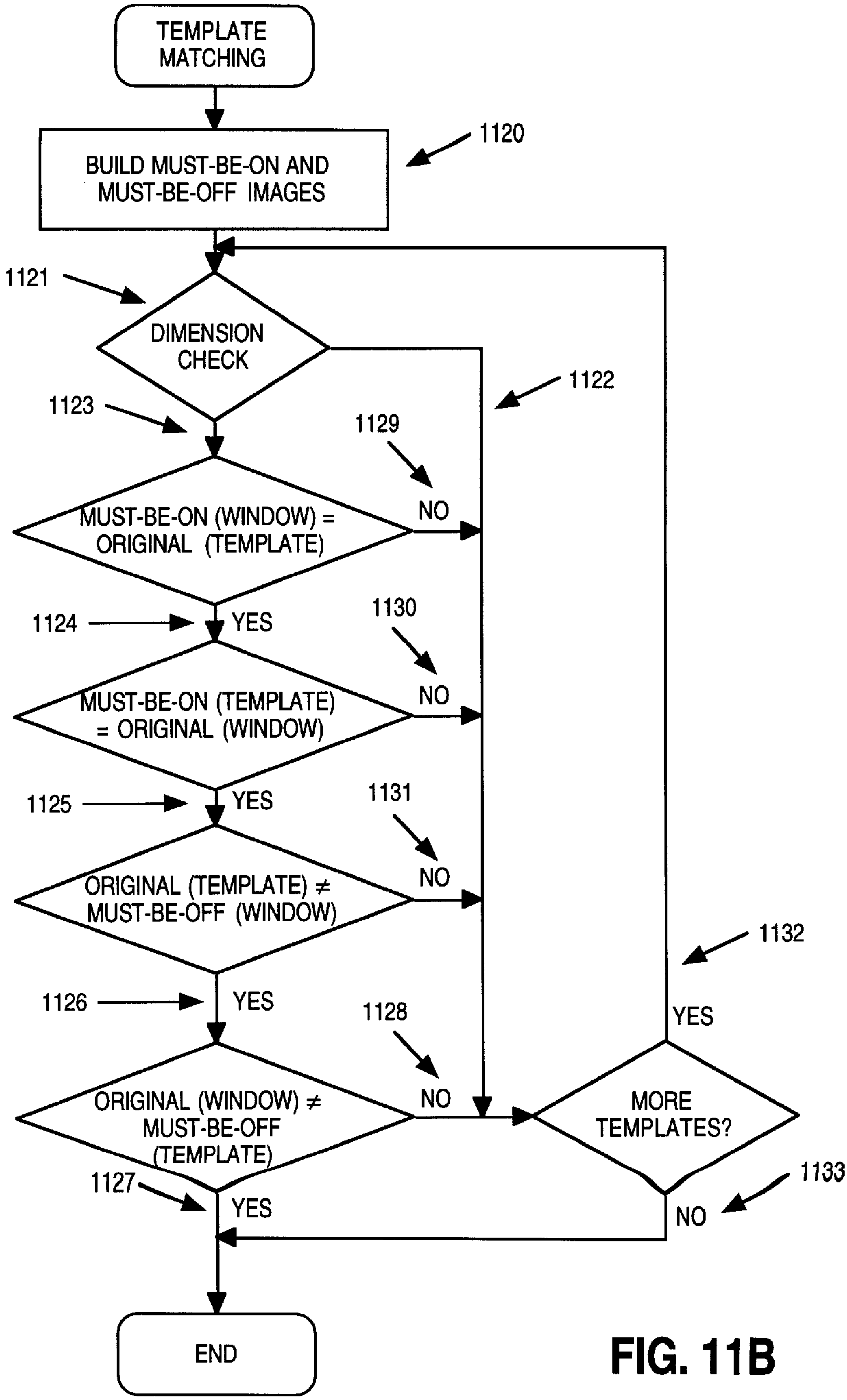

CHARACTER RECOGNITION
1101
TEMPLATE MATCHING
CHARACTER RECOGNIZED?
YES
1104
NO
1105
FEATURE ANALYSIS
END

FIG. 11A



TEMPLATE MATCHING
BUILD MUST-BE-ON AND MUST-BE-OFF IMAGES
1120
1121
DIMENSION CHECK
1122
1123
1129
NO
MUST-BE-ON (WINDOW) = ORIGINAL (TEMPLATE)
1124
YES
1130
NO
MUST-BE-ON (TEMPLATE) = ORIGINAL (WINDOW)
1125
YES
1131
NO
ORIGINAL (TEMPLATE) ≠ MUST-BE-OFF (WINDOW)
1132
1126
YES
1128
YES
NO
ORIGINAL (WINDOW) ≠ MUST-BE-OFF (TEMPLATE)
MORE TEMPLATES?
1127
1133
YES
NO
END


FIG. 11B

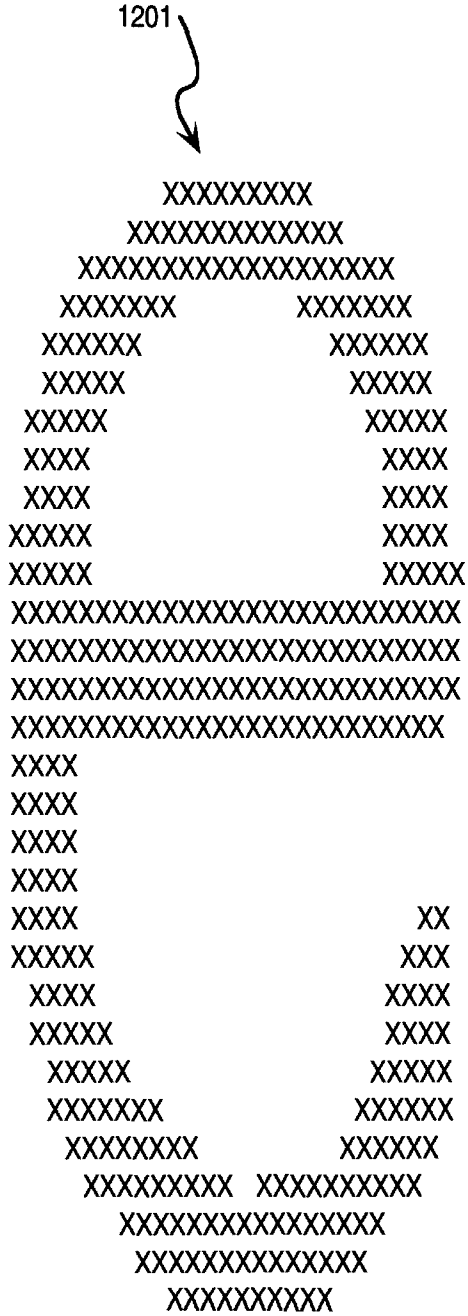



FIG. 12A

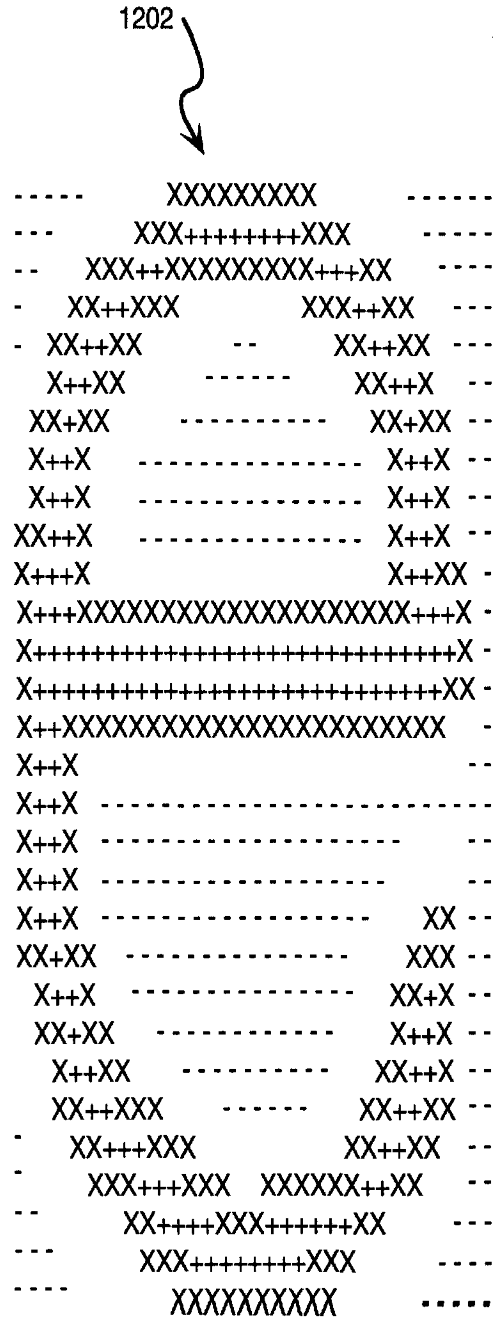



FIG. 12B

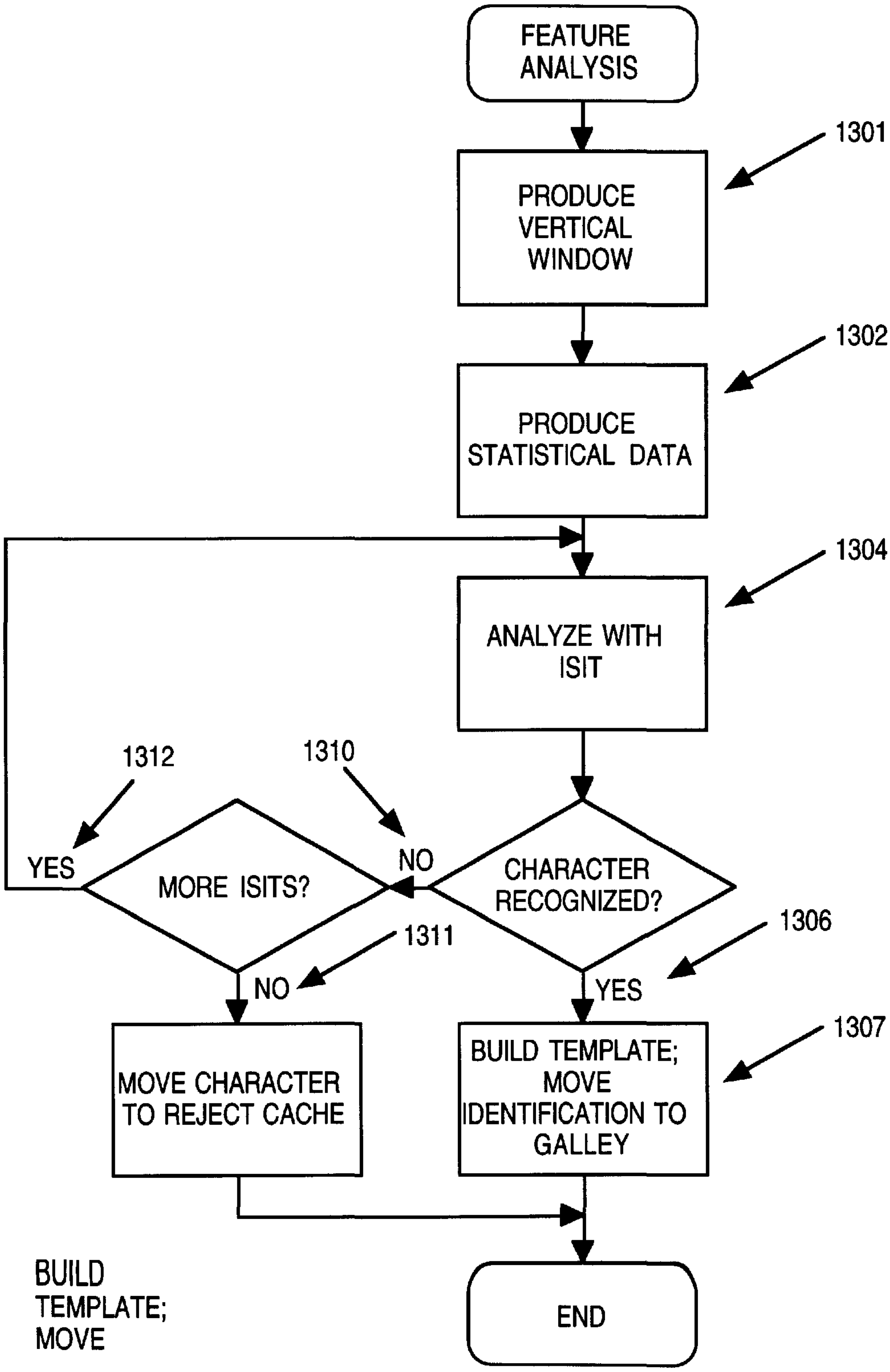

FEATURE ANALYSIS
1301
PRODUCE VERTICAL WINDOW
1302
PRODUCE STATISTICAL DATA
1304
ANALYZE WITH ISIT
1312
1310
YES
MORE ISITS?
NO
CHARACTER RECOGNIZED?
1311
1306
NO
YES
1307
MOVE CHARACTER TO REJECT CACHE
BUILD TEMPLATE; MOVE IDENTIFICATION TO GALLEY
BUILD TEMPLATE; MOVE
END

FIG. 13

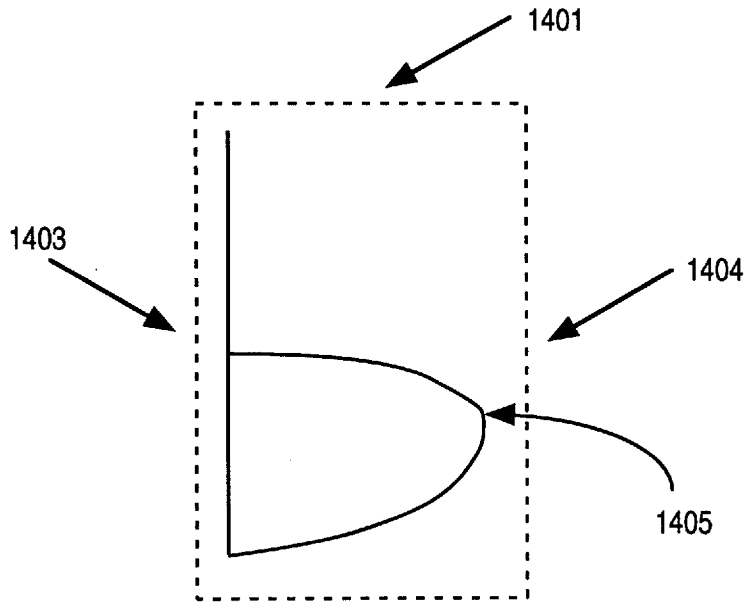



FIG. 14A

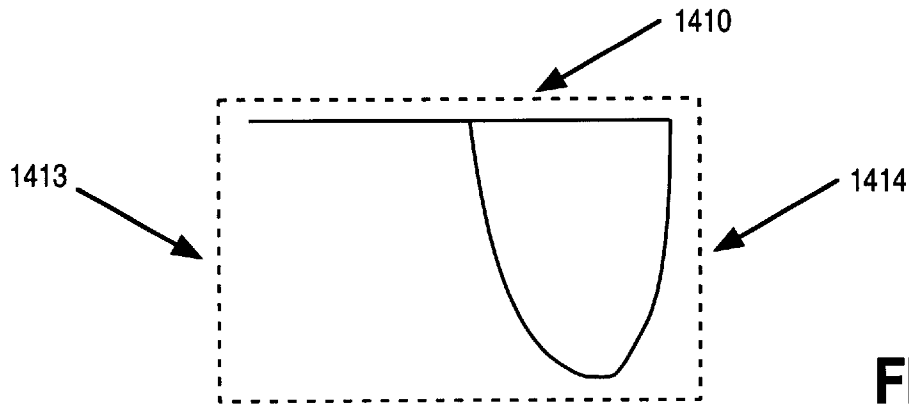



FIG. 14B

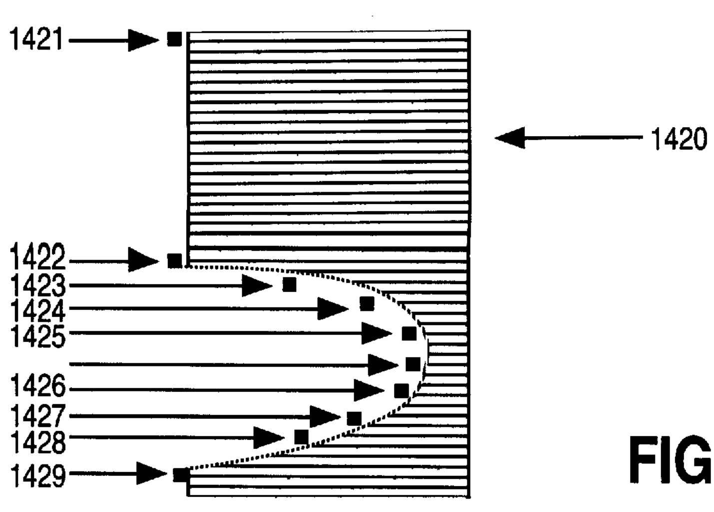



FIG. 14C

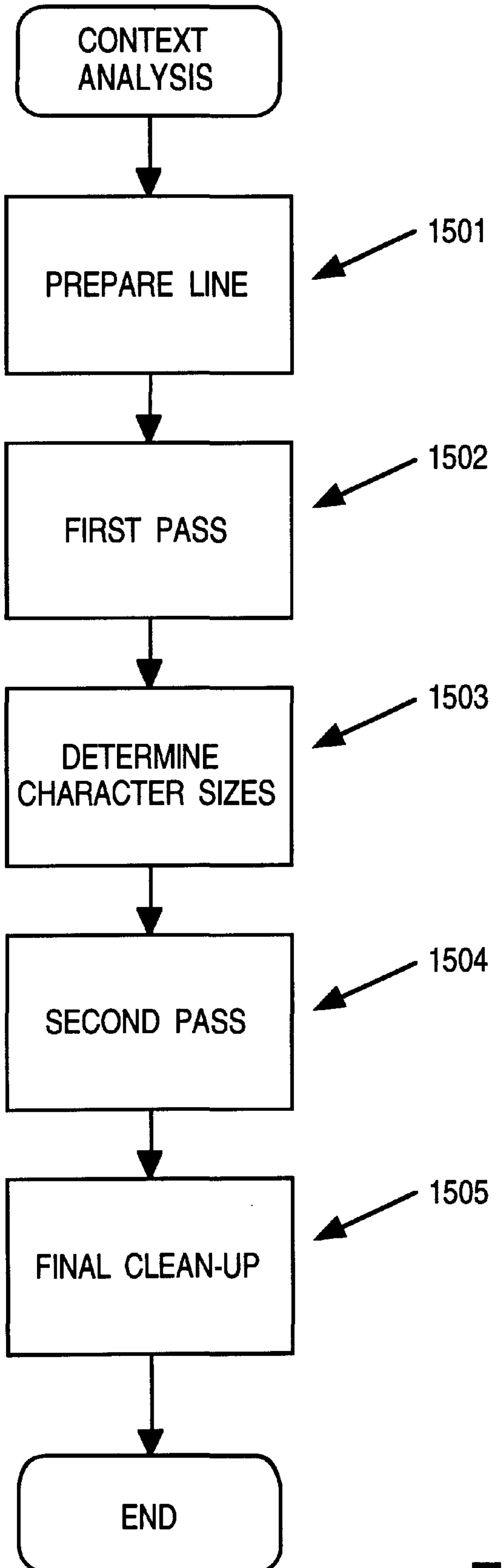

CONTEXT ANALYSIS
PREPARE LINE
1501
FIRST PASS
1502
DETERMINE CHARACTER SIZES
1503
SECOND PASS
1504
FINAL CLEAN-UP
1505
END

FIG. 15

# OPTICAL CHARACTER RECOGNITION METHOD AND APPARATUS

This is a continuation of application Ser. No. 07/914,120, filed Jul. 15, 1992 now abandoned, which is a divisional of application Ser. No. 07/799,549, filed Nov. 27, 1991, now U.S. Pat. No. 5,278,918, which is a continuation of application Ser. No. 07/230,847, filed Aug. 10, 1988 now U.S. Pat. No. 5,131,053.

## BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to the field of optical character recognition systems and, more specifically, to a method and apparatus for performing optical character recognition of printed text.

2. Prior Art

A number of optical character recognition (OCR) systems are known in the art. Typically, such systems comprise apparatus for scanning a page of printed text and performing a character recognition process on a bit-mapped image of the text. The characters may then be stored in a file on a computer system for processing by a word processor or the like.

Some known OCR systems comprise a hand held scanner for scanning a page. In such systems, the individual performing the scan, sweeps the hand held device over printed text on the page and will normally avoid scanning of graphics or non-text portions of the page. Normally, the page is scanned in the order in which it is normally read (i.e. the scan is performed down columns, scanning across columns from left to right).

Other known systems comprise a ruler apparatus which may be utilized for measuring or indicating portions of the text which are to be processed by the OCR system. Some of such systems are capable of discriminating graphic portions of the indicated page areas from text portions. However, such a system still requires manual intervention to mark off text in the order it is normally read and to mark off graphics portions.

Other systems utilize a registration mark to indicate the beginning of columns of text. These systems still require manual intervention to add registration marks.

Therefore, as one object of the present invention, it is desired to develop an optical character recognition method and apparatus which allows for scanning of a page of text without requiring manual intervention to mark off columns or otherwise indicate the normal order in which the text will be read. Further, it is an object of the present invention to develop an optical character recognition system which allows for a page of mixed text and non-text images to be scanned and for the system to recognize and distinguish between text and non-text for purposes of processing.

Known optical character recognition systems may be generally divided into two categories. Optical character recognition systems in the first category recognize either a single font or a limited number of fonts and their input is usually restricted to monospaced type of a specific point size. Optical character recognition systems in the second category are typically termed omnifont systems. Such systems are capable of recognizing a large number of typefaces in a wide range of point sizes, either monospaced or proportionally spaced. In general, optical character recognition systems which recognize a large number of typefaces are not capable of processing documents as quickly as systems which recognize a limited number of specific fonts.



It is another object of the present invention to develop an optical character recognition system which allows for recognition of any number of typefaces while still allowing for the rapid processing of documents.

These and other objects of the present invention will be described in more detail with reference to the Detailed Description of the Present Invention and the accompanying drawings.

## SUMMARY OF THE INVENTION

An optical character recognition method and apparatus is described. The system of the present invention comprises a scanning means for scanning a document and producing a bit-mapped image of the document. The scanning means is coupled with a computer system having a memory for storing the bit-mapped image and a processor for processing the bit-mapped image and providing as output, data representing the characters on the page.

The present invention discloses methods and apparatus to allow the page to be scanned and characters on the page to be recognized and output in an order which is logically the same order as that employed by a person reading the page. The present invention accomplishes this object by parsing the page into a plurality of blocks and outputting the blocks to a character recognition process in an order which is usually the logical reading order of the page.

The present invention further discloses a character recognition process comprising a combination of a template matching process and a feature analysis process. The feature analysis process allows characters to be recognized based on their shapes. Utilizing the feature analysis process of the present invention, it is possible to recognize characters in any number of different fonts. Further, by utilizing the template matching processes of the present invention in concert with the feature analysis processes, reasonable throughput of documents is achieved without the requirement of providing template libraries.

The present invention further discloses a context analysis process which completes the recognition process by iteratively resolving ambiguities of shape so as to minimize the number of typographic or semantic inconsistencies; residual inconsistencies are flagged as low confidence level identifications.

The present invention further discloses a number of techniques to provide for template matching such as producing representations of characters showing bits in a bit-mapped image of the character which must be off for the character to be recognized and, similarly, producing representations showing bits which must be on, This technique allows for recognition of characters within certain tolerances. The present invention further discloses a process for recognizing characters using character templates when characters are joined closely together.

Further, the present invention discloses use of a plurality of routines utilized in the feature analysis process. Each of these plurality of routines is designed for recognition of one character shape. The present invention discloses methods of describing characters with statistical information and fitting polygons onto different views of the shape of the character. Based on this statistical information and the polygons, the feature analysis processes are able to recognize shapes of characters.

The present invention further discloses a number of methods for distinguishing between text and graphics in a document. During processing of a document for purposes of character recognition, the present invention identifies areas

of a document which contain graphics and ignores such areas during the recognition process. The present invention discloses methods of measuring the relative texture of small areas of the document in order to determine whether the area contains text or graphics.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is an overall flow diagram of the optical character recognition process of the present invention.

FIG. **2**(*a*) is a flow diagram of an overall parsing process of the optical character recognition process of the present invention.

FIG. **2**(*b*) is a flow diagram illustrating a page parsing process of the present invention.

FIG. **2**(*c*) is a flow diagram illustrating a block parsing process of the present invention.

FIG. **2**(*d*) is a flow diagram illustrating a line parsing process of the present invention.

FIG. **3**(*a*) illustrates a portion of a page as may be processed by the apparatus and methods of the present invention.

FIG. **3**(*b*) illustrates a bit-mapped image of the portion of the page of FIG. **3**(*a*).

FIG. **4** is a flow diagram illustrating a method of the present invention for producing a text map array.

FIG. **5** is a flow diagram illustrating a method of the present invention for de-skewing a page.

FIG. **6** is a flow diagram illustrating a method of the present invention for locating paths in a scanned page.

FIG. **7** is a block diagram illustrating two data structures of the present invention.

FIG. **8** is a flow diagram illustrating a method of the present invention for locating blocks.

FIG. **9** is a first page image as may be scanned by the apparatus and methods of the present invention illustrating the first page image being divided into a plurality of blocks.

FIG. **10**(*a*) is second page image as may be scanned by the apparatus and methods of the present invention illustrating the second page image being divided into a plurality of blocks.

FIG. **10**(*b*) is a second illustration of the second page image as may be scanned by the apparatus and methods of the present invention.

FIG. **11**(*a*) is an overall flow diagram of a character recognition process as may be utilized by the present invention.

FIG. **11**(*b*) is a flow diagram of a template matching process as may be utilized by the present invention.

FIG. **12**(*a*) is an illustration of a bit-mapped image of a character as may be utilized by the present invention.

FIG. **12**(*b*) is an illustration of a second version of a bit-mapped image of a character as may be utilized by the present invention.

FIG. **13** is a flow diagram of a feature analysis process as may be utilized by the present invention.

FIG. **14**(*a*) is an illustration of a first character window as may be utilized by the present invention.

FIG. **14**(*b*) is an illustration of a second character window as may be utilized by the present invention.

FIG. **14**(*c*) illustrates a polygon fitting method of the present invention.

FIG. **15** is a flow diagram of a context analysis process as may be utilized by the present invention.



## DETAILED DESCRIPTION OF THE INVENTION

An optical character recognition system is described. In the following description, numerous specific details are set forth such as pixel densities, byte sizes, etc., in order to provide a thorough understanding of the present invention. It will be obvious, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known circuits, structures and techniques have not been shown in detail in order not to unnecessarily obscure the present invention.

Referring now to FIG. **1**, an overall flow diagram of the present invention is shown. Utilizing methods and apparatus of the present invention, first a page is scanned, block **101**. As will be described, the present invention is capable of isolating and translating the text on a page not only when the page contains only text but also when the page contains a combination of text and non-text areas. Further, the present invention requires no manual intervention to indicate the normal order of reading the text.

After scanning the page, the page is parsed, block **102**. Parsing of the page will be described in more detail below and may generally be divided into the functions of page parsing, block parsing and line parsing. After parsing the page, the shapes of individual characters are recognized through a character recognition process, block **103**. The present invention is capable of recognizing characters in any number of character fonts.

A process termed context analysis is employed to examine the relative sizes and positions of the shapes recognized during the character recognition process to divide the text into words and to resolve ambiguity of shape, block **104**.

Finally, the recognized characters are formatted for output, block **105**.

## SCANNER

The method and apparatus of the present invention, in its preferred embodiment, is designed to work in conjunction with commercially available microprocessors with 32-bit address spaces. Examples of such microprocessors are the Motorola 68020 and the Intel 80386 microprocessors.

It will be obvious to one of ordinary skill that the present invention may be practiced with any number of computer systems having a processor and memory available for practicing the described methods.

The scanner of the present invention may be any one of several known scanners presently commercially available or may comprise a yet unannounced scanner means. The present invention is designed to work in conjunction with a scanner which is capable of scanning a page of printed information and producing a bit-mapped image of the page. The present invention, in its preferred embodiment, seeks to utilize low-cost optical scanners and personal computer systems in order to provide a low-cost economical optical character recognition system.

## PARSING

After a page is scanned, a bit-mapped image of the age is stored in the memory of a computer system or in other means capable of storing such a bit-mapped image. With reference to FIG. **2**(*a*), the parsing process begins with page parsing, block **201**. The page parsing process acts on the bit-mapped image to divide the page into a plurality of blocks. The page parsing process attempts to divide the non-blank portions of the page into a plurality of blocks and

attempts to distinguish between text and non-text or graphics. The page parsing process attempts to ensure that any individual block contains either only text or only non-text. Blocks containing only non-text are eliminated from further processing.

Briefly, the page parsing process, block **201**, analyzes the texture of the scanned image to separate text from graphics and to detect the presence of columns and headlines. Based on a normalized two-dimensional measure of the local density of black-to-white transitions, areas on the page are characterized as either graphics, text or rulings. The areas are grouped into blocks by finding paths of white space surrounding areas of similar texture (e.g. having similar density of black to white transitions). If they are not separated by rulings or excessive white space, adjacent blocks with similar characteristics of texture and alignment are merged together. The page parsing process, block **201**, then analyzes the relative placement of blocks to determine which blocks overshadow other blocks in terms of line height, block width and vertical position on the page. The page parsing process can then construct a reasonable interpretation of the page layout. The final output of the page parsing process block **201**, is an ordered set of block descriptors that trace the normal reading sequence implied by the derived page layout.

A block parsing process block **202**, analyzes black-to-white transitions to compute degrees of skew and locate horizontal paths that divide a given text block into individual lines of text. The block parsing process, block **202**, detects vertical rulings and eliminates the ruling from further processing.

The line parsing process, block **203**, searches each line from left to right locating vertical paths of clear space. Such vertical paths typically separate individual words and characters from each other. The sections delimited by clear space are processed from the left end of each line and are passed to the recognition routines in a buffer.

A character recognition algorithm, block **103**, processes the buffered sections to attempt to recognize individual characters. As will be explained in more detail with reference to the character recognition processing section below, unrecognized line sections are run through a number of processing steps in an attempt to recognize characters. A "delining" process locates and erases from view of the character recognition process underlinings and horizontal rulings. An "unkerning" process isolates characters that are separated by sinuous paths of free space. A "blanding" process erases "pepper" noise. A "thinning" process thins extremely condensed characters to make them recognizable. A "patching" process mends slightly broken character shapes. Line sections which still remain unidentified after applying these processes are buffered in a reject cache for later processing by a "side matching" process and a context analysis process, block **104**.

## PAGE PARSING

Referring now to FIG. **2**(*b*), a flow diagram of the page parsing process is shown in more detail. The purpose of the page parsing process is to accept as input the bit-mapped page image and provide as output an ordered list of blocks of text. After a page is scanned, a bit-mapped image of the page is produced. From-this bit-mapped image, three arrays are produced, block **212**.

In the preferred embodiment, the page image is scanned at a resolution of 300 dots per inch, and the parsing process produces its arrays by analyzing every eighth scan line of the page image. It has been determined experimentally that sampling the page every eighth scan line (every 8/300th of an inch in the preferred embodiment) gives sufficient resolution to locate and categorize occupied areas on the page. Further, utilizing only every eighth scan line of the image substantially reduces processing time and storage requirements. However, it will be obvious to one of ordinary skill in the art that different samplings may be utilized without departure from the spirit and scope of the present invention.



Referring to FIG. **3**(*a*), an enlarged portion of a page **300** is shown. The portion of the page **300** shown represents an area of a page which may be covered in **40** scan lines of the preferred embodiment. Each of the squares, such as square **301**, represents an area 8 scan lines high and 8 bits wide. The area of the page shown **300** comprises an area 5 squares high and 4 squares wide.

Referring to FIG. **3**(*b*), a bit-mapped image of the same portion of a page is shown. FIG. **3**(*a*) and FIG. **3**(*b*) are illustrative of a problem encountered in optical character recognition systems. In such systems, a letter, such as the letter "O"**302** in FIG. **3**(*a*), may be represented by a relatively rough approximation of the letter "O", such as the representation **312** in FIG. **3**(*b*). Further, text and graphics may be intermixed on a single page. For example, graphics image **303** is shown mixed with the text of FIG. **3**(*a*). The equivalent bit-mapped area is shown at **313**. It is obvious to one of ordinary skill in the art that images may suffer from further problems of clarity and crispness as a result of being transformed to a digital bit-mapped image.

A first array generated by the page parsing process of the preferred embodiment is a horizontal population count array. Each element in this array contains a count of the number of one ("1") bits in four consecutive bytes (32 bits of a sampled scan line). Therefore, the area represented by each element of this array is 32 bits wide by 8 bits high. For example, with reference to FIG. **3**(*b*), the 32 bits at scan line **320** has 14 1-bits. Therefore, the 8-bit high by 32-bit wide area, contained in blocks **331**, **332**, **333** and **334**, is represented in the horizontal population count array by an element containing the value 14, the total count of one bits in scan line **320**. Scan line **321** has 0 1-bits, line **332** has 9 1-bits, line **323** has 5 1-bits and line **324** has 26 1-bits and each of these 8-bit high by 32-bit wide areas would be represented in the horizontal population count array with corresponding values.

A second array utilized by the preferred embodiment of the present invention is a vertical population count array. Each byte in the vertical population count array contains the total number of one ("1") bits in 4 bytes, one byte from each of four consecutive sampled scan lines. For example, an entry in the vertical population count array may represent bytes **340**, **341**, **342** and **343** and have a value of 4 (byte **340** has 2 1-bits, byte **341** has 0 1-bits, byte **342** has 2 1-bits and byte **343** has 0 1-bits). In the preferred embodiment, the vertical population count array comprises an array in which rows of the array represent columns of the bit mapped image and columns of the array represent rows of the bitmapped image. This leads to processing efficiencies in the implementation of the preferred embodiment.

A third array built by the page parsing process of the preferred embodiment is a horizontal phase change array. Each entry in this array represents 32 bits of a sampled scan line and, therefore, the horizontal phase change array has the same dimensions as the horizontal population count array. Each array element contains the count of horizontal phase changes (transitions between runs of 1s and runs of 0s) in the 32 bits. The area represented by each element of the hori-

zontal phase change array is 32 wide by 8 bits high. For example, the 32 bits at line **320** have 7 transitions from 1s to 0s or from 0s to 1s, the 32 bits at line **321** have no transitions, the 32 bits at line **322** have 8 transitions, the 32 bits at line **323** have 2 transitions and the 32 bits at line **324** have 2 transitions.

Based on the horizontal population count and horizontal phase change arrays, a text map array is produced, block **213**. Each element of the text map array represents an area 32 bits wide by 8 bits high.

Referring to FIG. **4**, a flow diagram of the process for constructing the text map array is illustrated. The text map array is built by a process which scans down each column of the horizontal population count array searching for a non-zero element, block **401**. A non-zero element in the horizontal population count array indicates the presence of text, graphics or rulings in the corresponding area of the bit-mapped image. After finding a non-zero element, the process searches down the column of the horizontal population count array for a zero element. For each element processed in a run of non-zero elements, the corresponding horizontal phase change counts are summed. The process also counts the total number of elements in a run of non-zero elements, block **402**.

If the number of rows in a run is greater than or equal to 2 and less than or equal to 12, branch **403**, and the summed total of the phase change counts is greater than or equal to 8 and less than or equal to 22, branch **404**, the corresponding area of the bit-mapped image has the textural signature of text. The text map array elements corresponding to each of the row elements in the run are set to a code indicating text is present, block **405**. In the preferred embodiment, these elements are set to a value of TXTX.

If the row count is not greater than or equal to 2 and less than or equal to 12, branch **406**, a check is made to determine if the row count is greater than or equal to 24, branch **407**. If the row count is greater than or equal to 24, the corresponding area of the bitmapped image contains a vertical ruling. The text map array elements corresponding to the horizontal population count elements in the run are set to a value indicating a vertical ruling is present, block **408**. In the preferred embodiment these elements are set to a value of TXVR.

If the row count is less than 2 or between 12 and 24, branch **409**, it is an indication that graphics are present. The corresponding text map array elements are set to a value indicating graphics are present, block **410**. In the preferred embodiment these elements are set to a value of TXGR.

If the row count lies between 2 and 12, but the phase change count is either less than 8 or greater than 22, branch **411**, it also indicates the presence of graphics. The corresponding text map array elements are set to the code indicating the corresponding bits of the bit-mapped page image contain graphics, block **412**.

If the end of the horizontal population count array has not been reached, branch **413**, column-wise processing continues, searching for the next non-zero element, block **401**. Otherwise, processing is completed, branch **414**.

It has been determined experimentally that the above process for determining whether areas of the bit-mapped image contain text, graphics or vertical rulings is a reasonably accurate mapping. Typically, characters in ordinary text occur in a wide range of heights normally occupying 2 to 12 sampled scan lines. Therefore, the above process tests for run row counts of non-zero elements in the range of 2 to 12 lines.



It has also been determined experimentally that although larger height characters have fewer phase changes in each sample than smaller height characters, the total number of phase changes in a given run of non-zero elements remains substantially constant over a character size range of approximately 4 to 24 points. Therefore, in the presently preferred embodiment a phase change count total for a run of non-zero elements in the horizontal population count array between 8 and 22 is an indicator of printed text.

The page parsing process then makes another pass to attempt to locate relatively large text, block **214**. The procedure for locating large text is essentially the same as the procedure described above for building the text map array except that the procedure for locating large text examines every fourth scan line and fourth element in the phase change array. Thus, the large text routine looks at the bit-mapped image with approximately ¼ of the resolution of the process for locating normal sized text and identifies text up to four times larger. In the preferred embodiment, the largest text found by this routine is 48 sample scan lines high. At 300 scan lines per inch, this is equivalent to text of 1.28 inches in height or about 92 points. Text map array cells corresponding to areas in the bit-mapped page image which are found to contain large text by the above process are set to a value to indicate they contain large text. In the preferred embodiment this value is TXTH.

The page parsing process attempts to locate and isolate blocks containing only text from blocks containing only white space, blocks containing graphics, or blocks containing vertical rulings. As part of this process, in the preferred embodiment, the page parsing process attempts to determine the approximate skew of the page and adjust the text map array and the vertical population count array for such skew, block **215**.

Referring to FIG. **5**, a flow diagram illustrating the method utilized by the preferred embodiment of the present invention for deskewing a page is illustrated. First, a skew total and a sample count variable are initialized to zero, block **501**.

The text map array is then scanned down each column searching for runs of text cells, block **502**. When a run of text cells is located a first variable, utilized when examining cells to the left of the present text cell (LEFT), is set to −1. A second variable, utilized when examining cells to the right of the present text cell (RIGHT), is set to 1, block **503**.

For each text cell following the first text cell in the run, cells to the left and right of the text cell in the text map array are examined. If the cell to the left is empty (i.e. the cell is not indicated as containing text, graphics or vertical rulings using the above-described TXTX, TXVR, TXTH or TXGR codes), the skew total variable is incremented by the current value of LEFT and the sample count variable is incremented by 1. If the cell is occupied (it contains TXTX, TXVR, TXTH or TXGR), LEFT is set to 1 and the skew total variable and sample count variable are not modified, block **504**. If the cell in the text map array to the right of the current cell is empty, the skew total variable is incremented by the value of RIGHT and the sample count variable is incremented by 1. If the cell is occupied, RIGHT is set to −1 and the skew total variable and sample count variable are not modified, block **505**.

If there are more cells in the run of text cells, the processing of blocks **504** and **505** is repeated for those cells, branch **506**. Otherwise, branch **507**, if there are more cells in the text map array they are scanned for another run of text cells, branch **508**. After the entire text map has been

examined, branch **509**, an approximate skew for the page is computed and the text map array and vertical population count array are shifted to compensate for the calculated skew, block **510**. The approximated skew is calculated by dividing the skew total variable by the sample count. This process yields an approximation of the skew for the page in number of rows per 4,096 columns. The text map and vertical population count arrays are then adjusted.

The above described process for adjusting for skew assumes that text on a page is generally arranged in horizontal lines on the page. The text map array of the present invention has good vertical resolution (to within 1/38th of an inch). When a page is skewed, blank cells will tend to appear in what should be text cells. The above-described method utilizes these assumptions to determine an approximate skew.

After adjusting for an approximate skew, horizontal and vertical paths of white space through the text are located, block **216**. The purpose of locating these paths is to isolate blocks of text. In the preferred embodiment, the deskewed text map array is examined for horizontal paths of white space (white space may be defined as cells in the text map array which are not coded with TXTX, TXVR, TXTH or TXGR). The method of the present invention considers an area to be a horizontal path if a white space area exists which is at least 8 pixels wide and 192 pixels long in the horizontal direction of the page. Similarly, vertical paths are located utilizing the text map array. The preferred embodiment of the present invention considers an area to be a vertical path if it is a white space area which is at least 16 pixels wide and 192 pixels long in the vertical direction of the page. The above-mentioned lengths and widths for horizontal and vertical paths have been determined experimentally by the Applicant to locate horizontal and vertical paths through printed text.

Referring now to FIG. **6**, a flow diagram illustrating a method utilized by the preferred embodiment of the present invention for locating paths through the text is illustrated. As a first step in locating paths, a process is utilized to cause top, bottom, left and right margins of the page to be effectively ignored, block **601**.

The preferred embodiment of the present invention accomplishes masking of the margins by creating a version of the vertical population count array in which each cell of the vertical population count array represents a 32×32 pixel square area of the original bit-mapped image. Effectively, each cell in this version of the vertical population count array represents four consecutive cells in a column of the original vertical population count array. The compressed version of the vertical population count array is then "smeared".

Smearing is a technique utilized by the present invention to shift an image in a number of directions and to preform a logical OR operation on the original bit-mapped image and the shifted bit-mapped image. Effectively, this technique expands areas containing text or graphics while narrowing the intervening spaces and margin areas. Remaining white margin areas are marked as unavailable for the path finding process in the vertical population count array. The smearing technique is described in more detail with reference to the description of character recognition, below.

An array called a path map array is then created for storing horizontal and vertical path information, block **602**. The path map array has the same resolution as the text map array. Each column represent 32 bits of a sampled scan line and each row represents a sample taken every 8th scan line of the image. In the path map array, the existence of a horizontal path is indicated by setting one of the bits in an entry in the array; the existence of a vertical path indicated by setting another one of the bits in the entry.



In the preferred embodiment, as a first step in populating the path map array, the left most and right most columns and the top and bottom rows of the path map array are set to indicate the presence of paths. This step ensures that a path that runs into the margin of a page will generate a block. When fully populated, the path map array outlines blocks of text and blocks of graphics with horizontal and vertical paths.

After the path map is created and the left most and right most columns and top and bottom rows of the path map are initialized, horizontal paths are generated, block **603**. Each row of the text map array is scanned for runs of empty entries (i.e. entries that are not set to TXTX, TXVR, TXTH or TXGR). If a run is at least a predetermined number of bytes in length, horizontal path bits in the corresponding entries in the path map array are set. In the currently preferred embodiment, the predetermined number of bytes is 6. Each horizontal path that is found utilizing this process is extended by one column to the left of the horizontal path and by one column to the right of the horizontal path. This ensures that the horizontal paths at block edges will meet vertical paths at the edge of the block even if there is either graphics or a headline within 32 pixels of the left or right edge of the block.

The text map array is then scanned to find vertical rulings (i.e. entries which are set to TXVR), block **604**. The corresponding entries in the path map array have a bit set to indicate that there is a vertical ruling at that location in the image. This bit will be examined in a later processing set in which relatively narrow paths are removed from the path map array.

Next, vertical paths are populated in the path map array, block **605**. Each row of the vertical population count array is scanned for runs of empty entries. If a run is at least a predetermined number of entries long, it is considered as a possible vertical path. In the preferred embodiment the predetermined number is six. If either end of a vertical path fails to coincide with a horizontal path, the vertical path is truncated until it does coincide at both ends with a horizontal path. Vertical paths always extend from an intersection with a horizontal path to an intersection with another horizontal path.

The path map is then examined and all vertical paths which are only one entry wide are removed unless the entry indicates the vertical path was set because of a corresponding vertical ruling in the image. The path map is then scanned again and portions of horizontal paths are erased where they do not begin and end at a vertical path, block **606**.

A block locating routine, block **217**, utilizes data structures previously built by the page parsing process, such as the page map array, and builds two new data structures; a block map array and a block list.

Referring now to FIG. **7**, the block map array **701**, comprises an array of the same dimensions as the text map array. Each 1-byte cell in the block map array, such as cell **702**, contains a block number for that cell. Cells which have not yet been designated as part of a block are designated in the preferred embodiment with a 0. In the preferred embodiment there can be a maximum of 255 blocks per page image. The block number, such as block number 1 at cell **702**, is a pointer into a block list **703**. Each entry in the block list

comprises information about each block such as coordinates of the block, column **705**, cell count information, column **706**, and distance to other blocks, column **707**. The coordinate information **705** comprises information on each block's top most, bottom most, left most and right most pixels. The cell count information **706** comprises information on the number of text cells, number of large text cells and number of graphic cells within the boundaries of the block.

Referring now to FIG. **8**, the process of locating blocks, block **217** from FIG. **2**, comprises the steps of a block finding routine first scanning through columns of the path map, block **801**. The block finding routine scans each cell of the path map for cells which have neither horizontal or vertical paths through them. After finding a cell which has neither a horizontal or vertical path through it, the block finding routine checks the corresponding cell in the block map array. If the corresponding cell in the block map array is unoccupied (i.e. current value is a 0), the block finding routine calls a block carving routine, block **802**. The block carving routine examines the path map array for the current cell to find vertical paths on the left and right sides of the current cell. The block carving routine then scans up rows of the path map array. For each row, the block carving routine locates vertical paths on the left and right of each cell above the current cell. When the block carving routine locates a cell whose left or right edge, as determined by the vertical paths, differs from the left or right edge, respectively, of the current cell by more than 1 cell, the block carving routine creates a horizontal path at that cell. The block carving routine similarly processes down the column of the current cell to determine the bottom row for the current block. This process produces a roughly rectangular block to be processed by later described methods of present invention.

After the block carving routine has determined the left, right, top and bottom edges of a roughly rectangular block of cells, a block statistics routine, block **803**, is utilized to mark other cells as belonging to the same block. The block statistics routine marks each cell in the block map array bounded by the left, right, top and bottom paths of the current block as belonging to the current block. Further, the block statistics routine counts the number of text cells and graphics cells in the current block by examining the text map array. The relative number of text cells versus graphics cells is used in determining whether the block is classified as a text or graphics block for later processing. The block statistics routine further gathers information on the average length of runs of occupied cells in the text map array for each column in the block. This information is used to determine the approximate height of the characters in the text blocks. The block statistics routine further histograms the number of runs of occupied cells in the columns of the text map array. The median point in this histogram approximates the number of lines of text in the block. The block statistics routine also computes the extreme left, right, top and bottom coordinates of the block. As discussed above, the above described coordinate, count and statistical information is stored in the block list, block **703**.

As previously described, the block list of the present invention allows for only 255 entries. The number of entries is limited in the preferred embodiment to allow for certain processing efficiencies. However, it will be obvious to one of ordinary skill in the art that a different number of entries may be utilized in the block list without departure from the spirit and scope of the present invention. In the preferred embodiment, if more than 255 blocks are found utilizing the above-described process, the process is repeated requiring a wider vertical path width to determine block boundaries. It will be further obvious that the present invention could employ a method of requiring wider horizontal paths as well as vertical paths.



After completing processing for a first block, processing continues with the block finding routine, block **801**, if more cells remain to be processed, branch **804**. After processing has been completed for all cells in the block map array, branch **806**, the process of locating blocks is completed.

Blocks are then grouped, block **218**. Blocks are grouped according to their location relative to one another, their contents (graphics or text) and their texture (font size and line spacing). Block grouping information is recorded in a group list. Each block in the block list is assigned a group number. The group number is used as an index into the group list.

For each block in the block list, the block map array is scanned above the block, below the block, to the left of the block and to the right of a block. Blocks which are of a different type or texture are not grouped together. Further, blocks which are more than a predetermined vertical distance apart or more than a predetermined horizontal distance apart are not grouped together.

Referring now to FIG. **9**, an example of grouping blocks is shown. For example, each of block **1**, block **6** and block **11** are grouped as group **1**, **901**. In the particular example, these blocks might comprise a heading on the page. A heading is often distinguished by having a larger type font than the rest of the text on the page. Because of the different font size, these blocks would be grouped together and not grouped with the remaining blocks on the page.

Block **2**, block **3**, block **4** and block **5** are grouped together as group **2**, **902**. The method of the present invention examines blocks adjacent to a current block to determine whether the horizontal distance H **903** is greater than a predetermined value. In the currently preferred embodiment, this predetermined value is 6 columns of cells. Since the horizontal distance H **903** between block **2** and block **7** is greater than the predetermined horizontal distance limit, blocks **7** and **8** are not grouped with group **1**.

Blocks **7** and **8** are grouped together as group **3**, **904**. Blocks **9** and **10** are grouped separate from blocks **7** and **8** and designated group **4**, **906**, because the vertical distance **905** between block **8** and block **9** exceeds a predetermined limit. In the preferred embodiment, the predetermined limit for vertical distance between blocks is **12** rows of cells. Blocks **12**, **13**, **14** and **15** are grouped together as group **5**, **907**. Block **16** is grouped separately as group **6**, **909**. Block **16** is not grouped with blocks **12**, **13**, **14** and **15** because of the existence of a graphics block **908**.

The present invention further discloses detecting edges of columns on a page by examining successive blocks down the page to determined if the left edge of each block is approximately lined up with the block below it and to determine if the block is vertically within a predetermined distance from its neighbor. If the block is not approximately lined with the block below it or the block is not a predetermined distance from its neighbor, it is assumed that the blocks are not in a column.

After the blocks are grouped, the block map array is rebuilt using the group numbers rather than the block numbers in each element in the block map array. This reduces subsequent processing requirements.

After completing grouping of blocks, blocks are arranged for output to the line parsing routines, block **219**. The purpose of arranging the groups for output is to attempt to output groups to the line parsing routines in the logical order

in which they would be read. Referring to FIG. **10**(*a*), a block diagram illustrating a page image consisting of 7 blocks of text is shown. The page image comprises a header area **1001** and 3 logic columns of text.

As a first step in arranging groups for output, vertically adjacent groups of blocks are located. Using the information regarding the location of vertically adjacent blocks, a tree is constructed linking the blocks. Each node in the tree represents a text or graphics block and contains pointers to up to 8 blocks above it. If there is more than **1** block above the current block, the pointers are arranged so that blocks are ordered from left to right. The root of the tree is at the bottom of the page. Each block is assigned a node number based on the left-to-right tree traversal order. Node **0** is assigned to the root.

As illustrated in FIG. **10**(*a*), node **1011** comprises a first vertical group **1002**. Node **2 1001**, Node **3 1013**, Node **4 1014** and Node **5 1015** may comprise a second vertical group **1003** being roughly adjacent to the first vertical group. Node **6 1016** and Node **0 1010** may comprise a third vertical group **1004** roughly adjacent to the second vertical group.

To determine the output order of the blocks, the tree is traversed from left to right and each branch of the tree is followed to its end traversing each subbranch from left to right. In general, a node at the end of a branch is output first and nodes from each branch of a subtree or are output before the root node for that subtree.

By way of example, with reference to FIG. **10**(*a*), to determine the normal output sequence for blocks **0–6** the tree is traversed from block **0** (the root) **1010** to the first node on the left, block **5 1015**. Nodes branching off of block **5 1015** are then traversed from left to right. Therefore, block **1 1011** is the next block examined. Since no blocks branch off of block **1 1011**, it is designated as the first block to be pruned from the tree and sent to the line parsing routines. The next node branching off of block **5 1015** is block **4 1014**. Therefore, block **4 1014** is processed next. Block **4 1014** has branches. Therefore, it is traversed and block **3 1013** is examined next. Likewise, branch **1013** is traversed and block **2 1001** is examined since it branches from block **3 1013**. As there are no other blocks which branch from block **2 1001**, block **2 1001** is the next block output to the line parsing routines. Block **3 1013**, having no more branches, is the next block to be output to the line parsing routines, followed by block **4 1014**. Since there are no more blocks branching off of block **5 1015**, block **5 1015** is the next block output to the line parsing routines. The root node **1010** continues to be traversed from left to right and block **6 1016** is processed. Again, since no blocks branch off of block **6 1016**, block **6 1016** is the next block sent to the line parsing routines. Finally, since no further blocks branch off of the root node **1010**, the root node is passed to the line parsing routines.

In processing blocks, those blocks designated as graphics blocks are included in the above described traversing and sorting process, however, graphics blocks are not passed to the line parsing routines.

When utilizing the above-described method for outputting blocks to the line parser, certain page layouts tend to obtain erroneous results. For example, in the page layout of FIG. **10**(*a*) blocks **2** and **3** may have been headers. In such a case, the logical reading order of the page would differ from the results produced by the above-described process. Therefore, the output order, in the preferred embodiment of the present invention of blocks, is rearranged by a recursively called routine. The routine is called for each node having more than one upward branch. For example, with reference to FIG. **10**(*a*), the routine would be called when processing block **5 1015** and the root block **1010**.

The recursive routine finds the top block of each branch. Starting with the left most branch, the routine examines the nodes on the next branch to the right. If the top of a node of the right branch is higher on the page image than the top of a node of the left branch (either the right node overlaps the left node or the bottom of the right node is above the top of the left node) the subtree on the right branch is grafted onto the left branch. This process repeats for each node which satisfies the above criteria.

For example, node **2 1001** and node **3 1013** are both above node **1 1011** so node **3 1013** is logically grafted, for purposes of output to the block parsing routines, to node **1 1011**. The tree after processing by this recursive routine is shown with reference to FIG. **10**(*b*). In FIG. **10**(*b*), the block numbers have been reassigned in left-to-right tree traversal order. The new block numbers indicate the order of output to the page parsing routines. The blocks are output to the page parsing routines starting with block **1 1001**, then block **2 1013**, block **3 1011**, block **4 1014**, block **5 1015**, block **6 1016** and then the root node, block **0 2020**. The logical connection between blocks has been modified at branch **1020**.

A number of final adjustments are made as part of the page parsing process. These adjustments include processes for further merging of adjacent blocks after the block output order is known, enlarging blocks into the white space surrounding the block, constructing a mock-up of the page image showing the placement and order of text blocks, reskewing the block map and path map arrays to reverse for the skew correction applied earlier in the process and building a block descriptor array comprising descriptive information about each block.

Importantly, the mock-up of the page image showing the placing and order of text blocks may be displayed on a device such as a graphics terminal by the user. The user may then alter the output order of the blocks if desired. The process allows for user correction of the output order of the blocks where the page parsing routines have made incorrect assumptions regarding the logical reading order of the page.

## BLOCK PARSING

Each block produced by the page parsing routine is passed, in the desired order, to the block parsing routine. The block parsing routine attempts to parse each block into individual lines of text. The block parsing routines utilize data structures built by the page parsing routines and the bit-mapped image of the input block to isolate individual lines and to add data to the block descriptive information in the block list. The data added to the block list comprises information identifying the leftmost column in the block, the width of the block, the height of the block, the number of lines in the block and the starting line number.

After receiving an input block, the block parsing routine computes the skew of the block, block **221** of FIG. **2**(*c*). The skew of the block is computed based on a detailed analysis of the phase change counts in the horizontal phase change array. Next, individual lines are isolated, block **222**, by examining the bit-mapped image for the block in conjunction with the phase change count analysis to determine the location of the possibly skewed horizontal white space which separates lines.

The block parsing process isolates and cuts between lines, block **223**, by locating horizontal paths of least resistance that most closely approximate the calculated skew. The

roughly horizontal path that separates lines of text may be interrupted by characters which have descenders or ascenders. For example, lowercase "g", "j", "p", "q", and "y", all have descenders. The block parsing routine cuts around such characters to ensure that the tails of such characters are left with the proper line when the line is passed to the line parsing routines.

Whenever the block parsing routine is unable to avoid or skirt around an obstacle within given tolerance levels, the block parsing routine measures the dimensions of the obstacle to determine whether the obstacle is a vertical ruling. If the obstacle is a vertical ruling, the obstacle is erased. If the obstacle is not a vertical ruling, the block parsing routines cut through the obstacle. Individual lines are isolated and buffered for processing by the the line parsing routines, block **224**.

## LINE PARSING

Each line output by the block parsing routines is used as input to the line parsing routines. The line parsing routines attempt to segment a line of text into individual characters. Referring to FIG. **2**(*d*), the line parsing routines of the preferred embodiment first find all columns in the line that have white space from the top of the line to the bottom of the line, block **231**.

The columns or segments having white space from the top of the line to the bottom of the line are then isolated and framed, block **232**. To frame segments, the line parsing process determines the left, right, top and bottom boundaries for pixel areas bounded by vertical white space. The boundaries are computed such that as little white space as possible is left around the pixel area.

If the resulting "framed" pixel area is wider than 64 pixels (the widest character width that may be processed by the preferred embodiment without reducing the resolution of the character) or the ratio of the width to the height of the framed pixel area is greater than 3 to 2, it is assumed that the framed pixel area comprises more than one character.

In such a case, the framed pixel area may comprise "kerned" characters. Kerned characters are characters which overlap each other, although they do not actually touch. In such a case, vertical white space may not exist between the characters. An example may be the letter "T" followed by "o". If a "T" is placed sufficiently close to an "o", no vertical white space occurs between them.

An unkerning process is applied to such relatively wide pixel frame areas. The unkerning process computes the leftmost clear path from the top of the line to the bottom. A clear path is defined as a list of connected vertical and horizontal segments that trace a clear path between two characters. If the unkerning processing is successful in finding a clear path, the left, right, top and bottom boundaries are recomputed for the pixel frame area to the left of the clear path. If the resulting frame is still wider than 64 pixels or still has a ratio of width to height of greater than 3 to 2 or if no clear path was found, an attempt is made to detect and remove underlining. If the process is successful in removing underlining, the line parsing process again attempts to find vertical white space.

After a pixel frame has been found which is not too wide, a character is considered to have been isolated, block **282** and a character window is created for the character recognition routines, block **233**. In the preferred embodiment of the present invention, a character window is a buffer area capable of holding characters up to 128 rows by 128 columns or 64 rows by 192 columns. If the pixel frame is too large to be placed into a character window, the pixel frame is scaled to allow it to fit into the window. The pixel frame is copied one row at a time into the window. If the framed pixel area was derived as a result of an unkerning process, the right border of the pixel area is defined as the clear path found during the unkerning process. Otherwise, each row copied into the window consists of the bits in corresponding rows of the pixel frame area between vertical white spaces (i.e. the presumed isolated character).



Any window which is built which has a column width of more than 128 pixels is put into a reject cache for later processing. Otherwise, the character recognition routines are called and the window is passed to the character recognition routines, block **234**. If a character is successfully processed by the character recognition routines, a code for its recognized shape is placed in a buffer area termed a "galley". Windows which are rejected by all of the character recognition routines are added to the reject cache for later processing.

## CHARACTER RECOGNITION

Referring now to FIG. **11**, the character recognition process comprises the steps of template matching, block **1101**, followed by feature analysis, block **1105**, if the character was not recognized by the template matching step, block **1101**.

The template matching process, block **1101**, attempts to match characters passed in windows from the line parsing process to templates of already identified characters. The feature analysis process, block **1105**, attempts to recognize features of characters which could not be matched to templates. Based on recognizing these features, the character shapes are identified.

As one inventive aspect of the present invention, characters which are recognized by the feature analysis process are used as templates for recognition of later occurring characters. In the preferred embodiment, a template cache is built for each new document. The template cache comprises characters which have been recognized through the feature analysis process for the current document. Characters in the template cache are utilized in the template matching process. By building the template cache based on characters recognized in the document through the feature recognition process, the present invention allows for recognition of any font which is recognizable with the feature analysis routines. By combining elements of feature analysis and template matching, the present invention offers the performance advantages of a template matching system with the omnifont characteristics of a feature analysis system.

## TEMPLATE MATCHING

The template cache comprises information for each available template for the current document. For each template, a header field contains identification information for that particular template. The header field also comprises offset pointers to three pixel patterns which are used by the template matching process.

The first pixel pattern is the original pattern of the character as recognized by the feature analysis process. In the preferred embodiment, the original pattern of x rows by y columns is stored as a two-dimensional array with the rows zero padded up to a word boundary.

A second pixel pattern, termed a "must-be-off" pattern, is derived from the original pattern. The must-be-off pattern comprises x+1 rows and y+1 columns with the rows also being zero padded to a word boundary.

A third pixel pattern, termed a "must-be-on" pattern, is derived from the original pattern and comprises x−1 rows by y+1 columns. The actual image of the must-be-on pattern only occupies x−2 rows by y−2 columns. However, in the preferred embodiment an area of x−1 rows by y+1 columns is reserved for processing convenience to ensure that the array is as wide as the must-be-off array.

As will be apparent from the following description of the template matching methods of the present invention, characters will be recognized by the template matching process when the characters are within certain predetermined tolerances of the templates. Allowance for characters being within predetermined tolerances is essential because two bit-mapped images of the same character rarely, if ever, match exactly. Image digitization is sensitive to differences in registration and the digitization process itself introduces edge noise. Furthermore, characters are often broken or otherwise deformed due to poor image quality when the characters were originally printed, when the substrate which is being scanned has been copied or when the substrate is optically scanned for the character recognition process. Therefore, a simple bit-by-bit comparison is not adequate for a recognition process. The must-be-on and must-be-off image patterns are utilized by the present invention to allow for some margin of difference between characters.

The must-be-off array comprises a pixel image indicating which pixels in the window must be off (i.e. be set to 0) in order for the window to be considered a match against the template. Referring to FIG. **12**(*a*) a bit-mapped image of the character e **1201** is illustrated. The x's in character e **1201** designate pixels which are on in the original image in the template cache.

In the preferred embodiment, the must-be-off image indicates pixels which are one or more pixels away from any on pixel in the original bit-mapped image. FIG. **12**(*b*) illustrates the character e **1202** showing the must-be-off pixels as dashes. In the preferred embodiment, the must-be-off pixels are computed by "smearing" the original pixel image. The smearing is accomplished by performing logical OR processes on each row of the original pixel image. Each row of the original pixel image is logically OR'ed with a copy of itself shifted left one bit. The result is logically OR'ed with the original row shifted right one bit. The result of this step is logically OR'ed with the result of applying the same step to the row immediately prior to, or above, the current row. The result of this step is logically OR'ed with the similarly processed row immediately subsequent to or below the current row. The result of this operation is an image of the original character in which every pixel of the original bit-mapped image is surrounded by eight on pixels; the pixel above, below, to the right, to the left, and the 4 pixels at 45°, 135°, 225° and 315° from the original pixel. This effectively adds a one pixel layer of girth to the original character image. The complement of the resultant image is saved as the must-be-off pattern.

The must-be-on image comprises a character image showing bits which must be on for a match to occur. In FIG. **12**(*b*), the character image of the character e **1202** is illustrated with plus signs showing pixels which must be one for a match to have occurred. To compute a must-be-on image, each row of the original pixel image is logically AND'ed with a copy of the row shifted one bit to the left. The result is logically AND'ed with a copy of the row shifted one bit to the right. That result is logically AND'ed with a similarly processed image of the row immediately above the current row. The result of the operation is then logically AND'ed with the similarly processed row immediately below the current row. When performing the logical AND operation with the rows above and below the current row, the above and below rows have been logically AND'ed with images of themselves as described in the first two steps of this process before performing the logical AND operations with the current row. This process produces an image of the original character in which only those pixels which were surrounded on all eight sides remain on. Effectively, this causes the image to be one pixel layer less thick than the original image.



Utilizing the must-be-on and must-be-off arrays for comparing the input characters to templates allows for tolerances to be used in performing the matching. Although the preferred embodiment allows for a one pixel tolerance in testing for matches, it will be obvious to one of ordinary skill in the art that alternative embodiments may allow for different tolerance levels. An alternative embodiment which allows for less stringent tolerances may lead to higher early match rates and, thus, faster processing. However, such an embodiment may have a greater rate of errors in identification of characters due to the less stringent tolerances.

Referring now to **11**(*b*), each time a new window comprising unidentified pixel information is received from the line parsing routine, must-be-on and must-be-off images are created for the unidentified image using the above-described processes, block **1120**. The unidentified image in the window is then compared against characters in the template cache. The templates are ordered, for purposes of comparison, in a most recently matched order. Each time a match occurs against a template a counter stored in the template header is incremented.

When a template is first created as a result of being recognized by the feature analysis routines, the template's match counter is set to 0. In the preferred embodiment, new templates (i.e. templates with a match count of 0) are inserted at the beginning of the template queue. When an unidentified image processed by the template matching routines matches a particular template, the particular template's match count is tested to determined if the match count is 0. If the match count is 0, the preferred embodiment of the present invention examines the image in the character window utilizing the feature analysis routines (described below) to obtain confirmation that the image in the character window is the same character as identified by the template. If the feature analysis routines confirm the template and the image in the character window is the same character, the match count is incremented. Otherwise, the process assumes that the template yields unreliable results and the template is discarded from further processing. The recognition of the image in the character window then continues by attempting to match the image in the character window against other templates in the template cache.

The first step in matching an image in a window against the template is to construct must-be-on and must-be-off arrays from the unidentified image, block **1120**. Next, a dimension check is made, block **1121**. Images which differ from a template in their height or width by more than one pixel can't match the template, branch **1122**. If the dimension check is passed, branch **1123**, the must-be-on array for the unidentified image in the window is compared against the original bit-mapped image of the template. If all of the pixels in the must-be-on array for the unidentified image are on in the original template, branch **1124**, a second test is performed.

The second test determines whether all of the pixels in the must-be-on array for the template are on in the unidentified

bit-mapped image in the window. If all such bits are on, branch **1125**, the original bitmapped image of the template is compared against the must-be-off array for the image in the window. All pixels indicated in the must-be-off array for the image in the window must be off in the original template for a match to have occurred. If this test is passed, branch **1126**, the unidentified bit-mapped image in the window is tested against the must-be-off array for the template. If all of the pixels indicated by the must-be-off array for the template are off in the bit-mapped image in the window, the template is judged a match and branch **1127** is taken.

As described above if the template has a match count of 0, the image in the character window is also analyzed with the feature analysis routines to confirm the identification. Otherwise, the identification code for the recognized shape is placed in the galley for later processing by the context analysis routines.

If any of branches **1122**, **1128**, **1129**, **1130** or **1131** are taken as a result of not passing the corresponding test described above and if more templates exist in the template cache, branch **1132**, the same series of tests are made against each successive template in the template cache until a match occurs or the cache is exhausted.

If there are no more templates in the template cache, branch **1133**, none of the current templates match the unidentified image. This occurs whenever the unidentified image differs in font, size or registration from all of the characters in the template cache. The no-match condition may also be a result of characters which, although of the same font and size, are not a close enough match to be within the "edge noise" tolerances of the template matching routines.

In any event, if the image is not recognized, branch **1104** of FIG. **11**(*a*), the feature analysis routines are called using the image in the character window as input, block **1105**.

## FEATURE ANALYSIS

The preferred embodiment of the present invention discloses use of a plurality of routines for analyzing the features of images passed as input to the feature analysis process to determine the category of the shape of an unidentified image in the character window. The plurality of routines comprises one routine for each unique species of shape in the standard character set. Each of these individual routines is capable of analyzing an image in a character window and providing as output an indication whether or not the image belongs to the generic shape category discriminated by the routine. The character recognition routines are exited when one of the routines responds with a positive indication that the image in the character window is the shape corresponding to that particular routine. If none of the feature analysis routines respond positively, the shape of the image in the character window remains unidentified. In such a case, further processing is performed to try to identify the shape of the image in the character window.

Each of the plurality of routines will be termed an "isit". The name "isit" is useful in describing the routines of the preferred embodiment because the routines determine whether a character in the character window is a particular character (e.g. "is it" an a). In the preferred embodiment isits exist for letters, numerals and special symbols such as commas, quotation marks, semicolons, etc. It will be obvious to one of ordinary skill in the art that the method of utilizing isits for determining whether an image in a character window is a particular character may be implemented for a wide number of alphabetic character sets. For example, isits may be implemented for cyrillic character sets such as the character set used in the slavic languages or other character sets such as the character sets for Hebrew or Arabic.



In the preferred embodiment, the isits distinguish characters based on their shape. Therefore, characters sharing the same topography are recognized by a single isit. For example, the letter lower case "p" and the letter upper case "P" are recognized by the same isit. The letters lower case "u" and upper case "U", lower case "s" and upper case "S", lower case "o", upper case "O" and zero "**0**", etc., are other examples of characters with the same or similar topography which would be recognized by the same isits. For each shape or topography, characteristics of the shape have been chosen and measured experimentally such that a particular isit can distinguish the shape of its character from the shape of other characters through a wide range of font styles.

In the preferred embodiment, an isit provides as output either an ASCII code for a particular character or a code indicating the character is recognized as belonging to a particular class of characters or a reject code indicating the character is not recognized. An output of the ASCII code for a particular character indicates the character's identification by the isit routine is unequivocal. The ASCII code returned is the standard ASCII code for the particular character. A code indicating the character belongs to a particular class of characters restricts any subsequent feature analysis to a particular set of isits.

Referring now to FIG. **13**, a flow diagram illustrating the feature analysis process of the present invention is shown. For each image received as input to the feature analysis process, statistical data from a horizontal window and a vertical window is used by the isits. The horizontal window is the original character window having the bit-mapped image of the character. For example, referring to FIG. **14**(*a*), the character "b" is shown in horizontal character window **1401**. A vertical window is derived from the horizontal window **1401**, block **1301**. The vertical window may be thought of as the image in the horizontal window **1401** laid on its side with each of the rows of the horizontal window reversed in position. For example, FIG. **14**(*b*) illustrates a vertical window for the image of the character "b" in vertical window **1410**.

Statistical information is produced by examining the bit-mapped images in the horizontal window **1401** and vertical window **1410**. The statistical information comprises profile data, polygon representations of the characters, phase change information, and counts of the number of on pixels in each row of the character.

The present invention's polygon fitting algorithms dampen the effects of noise in the images to be identified and greatly reduce of the volume of data that has to be processed by the feature-direction routines. Further, it has been determined that polygon representations of character images are consistent over a wide range of character sizes, e.g., the character "i" produces substantially the same polygon representations in a wide range of type fonts.

For each face of a character window, profile data and four polygons are derived. The faces of a character comprise the left and right sides **1403**, **1404**, **1413** and **1414** of the horizontal and vertical windows. The profile data comprises an array having one element for each row of the window. Each element holds a value, measured in columns, representing the distance from the edge of the frame to the first on pixel in that row. For example, referring to FIG. **14**(*a*) the first on pixel in each row of face **1403** would be the first on

pixel in the spine of the letter b. For face **1404**, the first on pixel for the top half of the letter b would be the first on pixel in the spine of the letter b. For the bottom half of face **1404**, the first on pixel for each row is the pixel located at the outside edge of the loop **1405** of the character b.

A first polygon is developed for each face. The polygon comprises a plurality of line segments fitted to a profile of the character viewed from the face. The line segments of the polygon are within a predetermined tolerance value of the profile of the character. For example, with reference to FIG. **14**(*c*), a profile **1420** of the character b viewed from face **1404** of FIG. **14**(*a*) is shown. The points on a polygon for describing this profile are shown at points **1421** through **1429**. In the preferred embodiment, there are a maximum of 16 points in the polygon used to described a particular profile and for each segment, its slope and the difference in slope from the prior segment are computed and saved. It will be obvious to one of ordinary skill in the art that a larger number of points may be used to described a polygon with a corresponding increase in processing and memory resources.

The polygon fitting algorithms determine the points such as points **1421** to **1429** on the profile image **1420**. The first step in the polygon fitting process is to assign polygon points **1421** and **1429** at each end of the polygon. A recursive routine is called using as inputs the end points of a line segment, such as points **1421** and **1429**, the profile image **1420** and a tolerance value. The tolerance value determines the "snugness" of the fit of the polygon to the profile image **1420**.

In the preferred embodiment, the tolerance (t) is measured in 128ths of a pixel and is computed based on the long and short dimensions of a window, according to the following formula:

Tolerance $(t)=(13/4)x+64$ if $x<28$;

and

$(t)=5x$ if $x>28$;

where

$x=(3*(\text{length of the long side})+(\text{length of the short side}))/4$.

The polygon fitting algorithm effectively draws a line between the end points **1421** and **1429** and locates the points furthest above and below the line (points **1422** and **1426**). If either extreme point lies beyond the allowed tolerance, it is incorporated into the polygon, thereby breaking the original line segment into subsegments. The algorithm continues, by recursively applying the same procedure to each subsegment until no raw data point lies further than the allowed tolerance form the nearest polygon segment. In the current example, both of the extreme points (**1422** and **1426**) are outside of the acceptable tolerance, so the original line segment is broken into three subsegments: **1421** to **1427**, **1422** to **1426**, and **1426** to **1429**. The algorithm continues by drawing the line segment between points **1421** and **1422**. This line segment has no points more than the allowed tolerance level above or below it, so it is not subdivided further. The algorithm then draws the line between points **1422** and **1426** and locates points furthest above and below the line. In this case, point **1425** is determined to be furthest above tolerance and no points lie below. This creates two new subsegments, **1422** to **1425** and **1425** to **1526**, which are refined recursively before the process examines the last subsegment that resulted from the first iteration of the process. The process eventually draws a line between point **1426** and **1429** and determines that point **1428** is furthest above the line and no points lie further than the tolerance value below it. The resulting subsegments are similarly refined by recursive application of the same procedure.



The process of iteratively creating subsegments by determining points that are maximally above and below the tolerance boundaries of existing segments continues until no raw data point lies more than the tolerance value from the nearest polygon segment of the profile **1420**.

A second polygon is comprised of line segments connecting points fitted to a representation of a face profile of a character called a shaded profile. The shaded profile is derived from a profile of the face of the polygon by traversing the polygon from the bottom row to the top row and subtracting the minimum X value encountered in traversing the polygon from the current outside point on the polygon. The same procedure is then repeated from the top of the polygon to the bottom. The effect is that the remaining non-zero X points in the profile represent areas which effectively would be shaded if the character were imagined as being illuminated by lights on both sides. The shaded polygon is used in detecting and analyzing openings into the interior of a character.

A third polygon is comprised of points on a profile on a face of a character which is developed by subtracting the shaded profile from the original profile. This is termed the filled polygon. Both the shaded and filled polygons have the same number of points with the same y coordinates as the original polygon.

Finally, a fourth polygon is developed which has a looser fit than the original polygon to the original profile. This is done by developing a polygon from the original polygon having fewer points (i.e. utilizing a larger tolerance value).

Data is developed for both the horizontal and vertical windows which counts the number of phase changes from white to black in each row of the respective windows. For example, the character "I" in the horizontal window is generally comprised of rows having one horizontal phase change from white to black and the character "H" is generally comprised of rows having two phase changes from white to black. The phase change information is used to generate statistics used by the isit routines.

A first statistic is developed showing the percentage of rows in which a particular phase change value occurs. For example, for the character "I", the value one (representing the occurrence of one phase change from white to black in the row) may occur close to 100% of the time. For the character "H", the value one may occur 5% of the time and the value 2 may occur 85% of the time. For the character "d", in the horizontal window, there may be 0 phase change counts of the value 0 indicating that every row has at least one pixel on. There may be approximately 55% of the rows having one phase change, these rows comprising the upper half of the character "d". The remaining approximately 45% of the rows have two phase changes, these rows comprising the loop of the character "d". There are no rows having 3 or 4 phase changes.

In the preferred embodiment, rows with more than 4 phase changes are counted with rows that have exactly 4 phase changes for purposes of calculation of these percentage values. The limit of 4 phase changes is developed from experimental data and observations developed by the Applicant. Typically, occurrences of more than 4 phase changes represent noise in the bitmapped image.

A second statistic is developed indicating where in the image of the character the average location of a particular

phase change count occurs. For example, for the character "I" for a phase change count value of 1, the "where" value may be close to 50. This indicates the average location of a line with a phase change count of 1 occurs approximately at the middle of the character. For the character "d" the where value for the phase change count for the value of 1 may be approximately 20, indicating the average of the row numbers containing one phase change is approximately 20% of the way down the character. The where value for a phase change count of 2 may be approximately 75, indicating the average of the row numbers containing 2 phase changes is approximately three-quarters of the way down the character. This is because the loop on a lower case "d" is at the bottom half of the character.

An array is also developed for both the horizontal and vertical windows comprising an element for each row of the respective windows showing a count of the number of on pixels in that row. Polygons are also fitted to these arrays.

The isits are used to individually analyze the character, block **1304**, until the character is recognized. A particular isit routine may utilize any one of a number of techniques for identifying whether a character is the character to be identified by the particular isit. In some cases, where a particular character's style varies significantly from type font to type font, a plurality of different techniques may be utilized to recognize the character. Typically, the isit routines analyze the shape of the faces of the character by utilizing the above described polygons and by utilizing the above described statistical information. The use of polygons to approximate characters tends to allow an isit routine to ignore small perturbations in the character.

A first technique utilized by isits to determine whether an image is the character to be identified by the particular isit is to examine the statistical information derived from the phase change data. For example, a particular isit may reject any character in which the percentage of rows having two changes from white to black exceeds 10%. An example of such an isit may be an isit for recognizing the character "I". In general, this method of examining statistical information allows most isits to eliminate 70 to 85% of all images input to them with a minimum of processing resources.

Another method of determining whether a particular character is a character to be recognized by a particular isit is to examine peaks in the polygons. For example, the letter upper case "F" is characterized by having two extreme peaks when examined from its right horizontal face. Isits may identify characters based on both the number of peaks and the relative position of such peaks. For example, an isit for recognizing the character "F" might reject the character "c" because the number and relative position of the peaks in one of its faces or in one of its population count arrays.

Certain characters are characterized by having a "spine" or "stroke" (i.e. the left face of a B, b, h, k, A, D, E, etc.). An isit for recognizing such characters may examine a character to find the longest single line segment in the character and look for characteristics such as the percentage of the length of the longest segment to the length of the character as a whole and the slope of the segment.

Another technique used by isits for identification of characters is to identify loops in the characters. A loop may be identified as a line having primarily two changes from white to black across each row of the loop. The isits identify the steepness of the curve of the loop, the relative symmetry of the curve and information about the corners of the loop. For example, the numeral "8" can often be distinguished from the letter "B" because of the difference in the corners.

As discussed above, after completing an analysis an isit returns either the ASCII code or the shape code for the particular image or information identifying the reason why the image was rejected. If the image is recognized, branch **1306**, a template is built as described above in connection with template matching, block **1307**. The identification code is then moved to the character galley.



Otherwise, the next isit is selected based on parameters such as the frequency of occurrence of the character represented by the isit and information obtained from prior isits showing reasons those isits rejected the character, branch **1310**. If there are no more isits, branch **1311**, the image in the character window is moved to the reject cache. Otherwise, branch **1312**, the next isit analyzes the character, block **1304**.

## SIDE MATCHING

In many cases, a character can not be recognized by either the template matching or feature analysis routines because the character is "stuck" to a character adjacent to it. This occurs, for example, when characters are spaced too closely together for the quality of the printing process. As one particular instance, such "stuck" together characters may occur when the ink bleeds in a document composed of proportionally spaced text.

The present invention discloses a method termed "side matching" for identifying such stuck characters. When analyzing a character window using side matching, templates are effectively laid on the sides of the character window to determine if a match would occur with that template if all pixels in the "shadow" of that template were ignored. The side matching methods of the present invention utilize the same template comparison algorithms as discussed above in reference to template matching but the side matching methods effectively ignore the images right or left sides.

For example, if a character window comprises the character "ite", all effectively stuck together, the side matching process attempts to lay templates on the left and right sides of the character window. Each of the templates would be aligned with the left side of the character window when looking for a match on the left and pixels to right of the on pixels in the template would be ignored. In the current example, laying a template for the character "i" on the left side of the character window could produce a match. If it did, the ASCII code for an "i" would be registered in the galley, and the side matching process would remove the pixels representing the character "i" from the window. The process would continue by attempting side matching on the remaining pixels in the window. In this case, a match might occur when using a template for the character "t".

The side matching method of the present invention may be applied from either the right or left side of a character window and is applied at all possible vertical registrations. If side matching from the left side of the character window leaves unrecognized characters, side matching is attempted from the right side.

Since some letter shapes are subsets of others (e.g., r-n-m or c-o-d), the templates are ranked and ordered by size prior to using the templates in side matching. Some templates (e.g., a period) are excluded from the side matching process because such templates would produces matches on virtually all images.

In the preferred embodiment, side matching is attempted on character windows in the reject cache after completion of processing of all characters in the document. This allows for a relatively large library of templates to be been created and increases chances for successful identification of characters with the side matching technique.

## CONTEXT ANALYSIS

The character recognition process identifies characters by their shape. However the shape of the character alone may not be deterministic of what the character is. For example, a lower case "s" may not be distinguishable from an upper case "S". As another example, an apostrophe is not distinguishable from a comma based strictly on its shape. The context analysis routine accepts as input and utilizes as output the galley of character codes created by the character recognition routines. Context analysis is performed on one line of the page at a time to attempt to resolve ambiguities. Referring now to FIG. **15**, the process utilized by the preferred embodiment of the present invention is illustrated.

The preferred embodiment of the present invention comprises a data base of characteristic attributes of various characters. These attributes may include information on whether the character is typically found entirely above the baseline of the character line or whether a tail or other portion of the character typically extends below the baseline. The data base also comprises information regarding the relative size of characters which are normally ambiguous when identified by shape alone. For example, the data base may comprises information for distinguishing between a upper case "S" and a lower case "s" based on the expected relative size.

Each line from the page is copied from the galley into a buffer to prepare the line for further processing, block **1501**. During the process of copying a line to the buffer, values are assigned to characters from the character attribute data base such as information on whether the character sits above the baseline and whether the relative size of the character indicates whether it is upper or lower case. Spacing between words is also determined at this point by constructing and analyzing a histogram of the distances between letters. Importantly, as ambiguities are resolved for one character, the present invention utilizes information gained from resolving such ambiguities to assist in resolving ambiguities about neighboring characters.

The base line is also adjusted for skew. In the preferred embodiment, the skew may be adjusted by examining the expected baseline for each individual character and attempting to adjust the baseline for the entire line based on this information. If, however, the values for the baseline disagree significantly from character to character or from word to word, a character near each end of the line is found which is reliably known to sit on the baseline (e.g. the character "B" is known to sit on the baseline, the character "Q" can not be reliably predicted to sit exactly on the baseline because in some fonts its tail may extend below the baseline). An adjusted base line is then determined by effectively drawing a line connecting the bottom of these two characters near each end of the line.

The typical heights of upper and lower case letters in the line are determined by preparing histogram information showing the heights of non-ambiguous characters. Normally, such a histogram will show two levels of peaks, a first level corresponding to lower case characters and a second level corresponding to upper case characters.

Certain character types such as underlines are moved to the end of the buffer area. This allows these characters to be effectively ignored during the majority of the context analysis processing. Such characters are restored to their previous positions in the line near the completion of the character analysis processing.

Finally, a histogram type analysis of the width of white space between adjacent characters on the line is prepared. In the preferred embodiment, punctuation characters are not included in the histogram analysis unless the line consists entirely of punctuation lines. Typically, the histogram has two peak levels. The first peak level is assumed to represent character spacing between letters within words and the second peak level is assumed to represent spacing between words. If there not two distinct peak levels, words spacing is assumed to be some midpoint between peak levels. If there are not at least two peaks, an arbitrary word spacing point is determined based on the average height of characters in the line.

After preparing the line for context analysis, block **1501**, a first pass is made through each character on the line, block **1502**, to attempt to resolve ambiguities. This first pass looks at such characteristics as the relative height of characters in each word, the positions relative to the baseline, etc. For cases in which there is an ambiguity in determining whether a character is a numeral or a letter, the character is analyzed in relation to the other characters that neighbor it to determine whether they are numeric or alphabetic. Determining character ambiguities is an iterative process in which information which is known about neighboring characters is used in analyzing a particular character. After all characters in a word have been examined, consistency checks are performed. If one or more characters are found to have inconsistent characteristics, all characters in the word are flagged as being possibly wrongly interpreted. A second pass of the context analysis routine is intended to correct the interpretation.

After completing the first pass context analysis for each line in the galley, the context analysis routine attempts to assign a font identification number to each character in the galley and to determine character sizes for each font, block **1503**. Font identification is propagated through the galley by tracing through all of the characters which were identified by templates. All of the characters that matched a particular template are linked together in a linked list having a root pointing to a particular template. Based on these linked lists of characters, words are assigned to fonts on the premise that words containing characters identified by the same template are of the same font. This is a lengthy, iterative process. Histogram information is then prepared detailing the height of the upper and lower case characters for each font.

A second pass is then made through each line in the galley, block **1504**. Words which were flagged during the first pass as having inconsistencies are again analyzed to determine which characters are incorrect. The second pass checks such items as baseline uniformity, character size uniformity, alphabetic/numeric context, etc.

Finally, a series of miscellaneous clean-up routines are utilized, block **1505**. Such things as punctuation are checked to ensure reasonable size and position. For example, a very large punctuation mark is probably an unrecognized character and is flagged as such. Periods and relatively wide commas or apostrophes which are located above the baseline or below the top of lower case character may be short dashes or bullets. These characters are assigned the ASCII code for a dash. When recognition is attempted on a bit-mapped image which is not printed text, such as a signature or a logo, the result is typically a string comprising unrecognized characters and a set of other characters such as punctuation and dashes. Such strings are collapsed into a single unrecognized character. Successive single quotes are combined into double quotation marks.

The context analysis routine also attempts to merge characters which may have been split by the character recogni-

tion routines. For example, two characters which may have been recognized by the character recognition routines as an open parenthesis "(" followed by a close parenthesis ")" may actually be the letter "o". The context analysis routines attempt to merge such split characters by recognizing the proximity of particular character pairs.

Other characters, such as underlines, are analyzed to determine whether they are within a predetermined distance from the baseline. If the underline is more than the predetermined distance or if its edges do not coincide with the word boundaries, it is considered to be a ruling rather than an underline. Otherwise, the characters above the underline are flagged as being underlined. The predetermined distance is determined based on the relative context of the line including such factors as the size of the characters.

The context analysis routines attempt to identify unidentified characters by merging together broken pieces of characters, resubmitting characters to the character recognition routines allowing less stringent constraints for the recognition, etc.

The output from the context analysis routines is the completed scanned page having ASCII character representations for characters on the page in the normal reading sequence of the characters.

Thus, an optical character recognition method and apparatus has been described.

We claim:

**1**. A method for optically recognizing characters on a medium and storing a template of said recognized characters in a template cache, said template cache for recognition of subsequent characters on said medium, said method comprising the steps of:

(a) analyzing a first shape characteristic of a first character with a first shape characteristic analyzing process;

(b) analyzing a second shape characteristic of said first character with a second shape characteristic analyzing process;

(c) identifying said first character based upon the first and second shape characteristics;

(d) generating a template of said identified first character, and

(e) storing said template in said template cache.

**2**. The method as recited in claim **1** wherein said step of generating a template of said identified first character is further comprised of the steps of:

(a) generating a must be off pixel pattern, and

(b) generating a must be on pixel pattern.

**3**. The method as recited in claim **1** wherein said step of storing said template in said template cache is further comprised of the steps of:

(a) storing a pixel pattern of said identified character in said template cache;

(b) storing said must be off pixel pattern in said template cache;

(c) storing said must be on pixel pattern in said template cache, and

(d) storing a header in said template cache, said header for associating said pixel pattern, said must be off pixel pattern and said must be on pixel pattern.

**4**. In an optical character recognition system having a feature analysis process for identifying an unknown character, said optical character recognition system for identifying characters in a medium, a method for constructing a template library for use while processing said medium, said method comprising the steps of:

(a) identifying said unknown character with said feature analysis process;

(b) building a template for said unknown character subsequent to having identified said unknown character; and

(c) storing said template in said template library.

**5**. The method as recited by claim **4** wherein said template library is created for each new medium presented to said optical character recognition system.

**6**. The method as recited in claim **4** wherein prior to said step of identifying said unknown character with said feature analysis process, performing said step of determining that said unknown character does not match a template in said template library.

**7**. In a character recognition system having template matching as a first character recognition process, a method for generating a template cache for recognizing characters on a medium being processed, said method comprising the steps of:

a) providing a first pixel pattern of an unrecognized character on said medium being processed;

b) determining said unrecognized character on said medium being processed does not match a template in said template cache;

c) recognizing said unrecognized character utilizing a second character recognition process to create a recognized character;

d) building a template of said recognized character; and

e) adding said template of said recognized character to said template cache.

**8**. The method as recited in claim **7** wherein said step of building a template of said recognized character is further comprised of the steps of:

a) generating a second pixel pattern corresponding to pixels which must be off;

b) generating a third pixel pattern corresponding to pixels which must be on; and

c) generating a header for associating said first pixel pattern, said second pixel pattern and said third pixel pattern.

**9**. The method as recited in claim **8** wherein said second character recognition process is a feature analysis process.

**10**. A system for optically scanning a medium to recognize characters thereon, said system comprising:

scanning means for providing as output a bit-mapped image of said medium;

a memory means coupled with said scanning means, said memory means for storing said bit-mapped image;

a parsing means coupled to said memory means, said parsing means for extracting unknown characters from said bit-mapped image;

a template cache memory for storing templates of recognized characters;

a template character recognition means coupled to said template cache and said parsing means, said template character recognition means for recognizing characters by comparing unknown characters to templates in said template cache;

a feature analysis recognition means coupled to said template cache and said memory means said feature analysis recognition means for recognizing characters not recognized by said template character recognition means; and

template generation means coupled to said feature analysis recognition means and said template cache memory,

said template generation means for generating a template of recognized character and storing in said template cache.

**11**. The system as recited in claim **10** wherein a template is comprised of:

a pixel pattern of a recognized character;

a must-be on pixel pattern of said recognized character; and

a must-be off pixel pattern of said recognized character.

**12**. The system as recited in claim **11** wherein said template generation means is further comprised of:

means for generating a must-be on bit pattern from a bit pattern of said recognized character; and

means for generating a must-be off bit pattern from said bit pattern of said recognized character.

**13**. A method of creating a character template using an image, said image including a representation of a character, said method including the steps of:

a) accessing said representation;

b) analyzing said representation using a first recognition process;

c) analyzing said scanned representation using a second recognition process, if said character is not recognized using said first recognition process;

d) generating a character template, if said character is recognized using said second recognition process, and

e) storing said character template, if a character template has been generated.

**14**. The method of claim **13** wherein said first recoginition process includes template matching and said second process includes feature analysis.

**15**. The method of claim **14** wherein said character template includes a first pixel pattern, a second pixel pattern, and a third pixel pattern, said first pixel pattern corresponds to said representation, said second pixel pattern includes a must-be-off pattern, and said third pixel pattern includes a must-be-on pattern.

**16**. A method of recognizing a first character in a character recognition system, said character recognition system



including an image, said image including a first image representation corresponding to said first character, said method comprising the steps of:

a) accessing said first image representation;

b) applying a template matching process to said first image representation;

c) applying a feature analysis process to said first image representation, if said template matching process does not successfully identify said first image represenation as corresponding to said first character,

d) identifying said first image representation as corresponding to said first character, and

e) including a first character template corresponding to said first image representation for use in subsequent template matching processing.

**17**. The method of claim **16** wherein said image includes a second image representation corresponding to said first charater, and said method further comprises the steps of:

f) accessing said second image representation, and

g) applying said templating matching process, including said first character template, to said second image representation.

**18**. A system for recognizing a character represented on an image, said system comprising:

a) a memory for storing said image and a set of character templates, and

b) a processor being coupled to said memory, said processor for accessing said image and generating a first representation of a portion of said image, said portion including a representation of said character, said processor further for applying a templating matching process to said portion, said processor further for applying a feature analysis process to said portion if said template matching process does not identify said portion as corresponding to said character, said processor further for generating a character template from said portion if said feature analysis identifies said portion as corresponding to said character.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

# CERTIFICATE OF CORRECTION

| | | |
|---|---|---|
| PATENT NO. | : 6,038,342 | Page 1 of 1 |
| APPLICATION NO. | : 08/114608 | |
| DATED | : March 14, 2000 | |
| INVENTOR(S) | : Phillip Bernzott et al. | |

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Delete Claims 1-3 at col. 27, Lines 28-61.

Signed and Sealed this
Fourteenth Day of August, 2012

David J. Kappos

David J. Kappos
*Director of the United States Patent and Trademark Office*

UNITED STATES PATENT AND TRADEMARK OFFICE

# CERTIFICATE OF CORRECTION



PATENT NO. : 6,038,342
APPLICATION NO. : 08/114608
DATED : March 14, 2000
INVENTOR(S) : Phillip Bernzott et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Delete the title page and substitute therefore the attached title page showing the corrected number of claims in patent.

Delete Claims 1-3 at col. 27, Lines 28-61.

This certificate supersedes the Certificate of Correction issued August 14, 2012.

Signed and Sealed this
Fourth Day of September, 2012

David J. Kappos

David J. Kappos
*Director of the United States Patent and Trademark Office*

# United States Patent [19]

**Bernzott et al.**

[11] **Patent Number: 6,038,342**

[45] **Date of Patent: Mar. 14, 2000**

[54] **OPTICAL CHARACTER RECOGNITION METHOD AND APPARATUS**

[75] Inventors: **Phillip Bernzott**; **John Dilworth**; **David George**, all of Oakland; **Bryan Higgins**; **Jeremy Knight**, both of Berkeley, all of Calif.

[73] Assignee: **Caere Corporation**, Los Gatos, Calif.

[21] Appl. No.: **08/114,608**

[22] Filed: **Aug. 31, 1993**

**Related U.S. Application Data**

[60] Continuation of application No. 07/914,120, Jul. 15, 1992, abandoned, which is a division of application No. 07/799,549, Nov. 27, 1991, Pat. No. 5,278,918, which is a continuation of application No. 07/230,847, Aug. 10, 1988, Pat. No. 5,131,053.

[51] **Int. Cl.**[7] .......... **G06K 9/34**

[52] **U.S. Cl.** .......... **382/173**

[58] **Field of Search** .......... 382/14, 15, 25, 382/30, 34, 33

[56] **References Cited**

U.S. PATENT DOCUMENTS

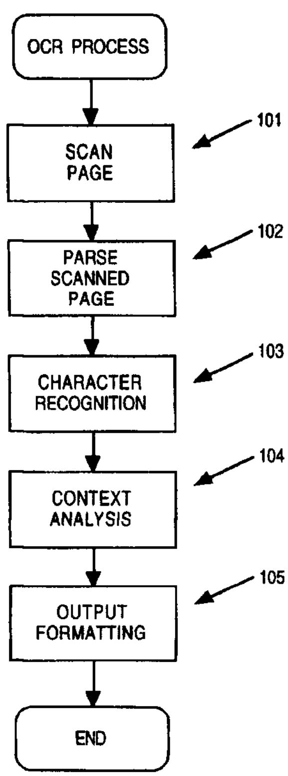

| | | | |
|---|---|---|---|
| 3,104,369 | 9/1963 | Rabinow et al. | 382/30 |
| 3,713,099 | 1/1973 | Henstreet | 340/149 |
| 3,713,100 | 1/1973 | Henstreet | 340/149 |
| 3,717,848 | 2/1973 | Irvin et al. | 382/30 |
| 3,873,972 | 3/1975 | Levine | 382/14 |
| 3,930,231 | 12/1975 | Henrichan, Jr. et al. | 382/30 |
| 4,027,284 | 5/1977 | Hoshino et al. | 382/30 |
| 4,499,596 | 2/1985 | Casey et al. | 382/15 |
| 4,672,678 | 6/1987 | Koezuka et al. | 382/30 |
| 4,850,026 | 7/1989 | Jeng et al. | 380/14 |
| 4,887,304 | 12/1989 | Terzian | 382/30 |
| 4,944,022 | 7/1990 | Yasujima et al. | 382/14 |
| 5,014,327 | 5/1991 | Potter et al. | 382/14 |


*Primary Examiner*—Jose L. Couso
*Attorney, Agent, or Firm*—Blakely, Sokoloff, Taylor & Zafman LLP

[57] **ABSTRACT**

A system for recognition of characters on a medium. The system includes a scanner for scanning a medium such as a page of printed text and graphics and producing a bit-mapped representation of the page. The bit-mapped representation of the page is then stored in a memory means such as the memory of a computer system. A processor processes the bit-mapped image to produce an output comprising coded character representations of the text on the page. The present invention discloses parsing a page to allow for production of the output characters in a logical sequence, a combination of feature detection methods and template matching methods for recognition of characters and a number of methods for feature detection such as use of statistical data and polygon fitting.

**15 Claims, 21 Drawing Sheets**



US006038342C1

(12) EX PARTE REEXAMINATION CERTIFICATE (9211th)

# United States Patent

Bernzott et al.

(10) Number: US 6,038,342 C1

(45) Certificate Issued: Aug. 21, 2012

(54) **OPTICAL CHARACTER RECOGNITION METHOD AND APPARATUS**

(75) Inventors: **Phillip Bernzott**, Oakland, CA (US); **John Dilworth**, Oakland, CA (US); **David George**, Oakland, CA (US); **Bryan Higgins**, Berkeley, CA (US); **Jeremy Knight**, Berkeley, CA (US)

(73) Assignee: **USB AG. Stamford Branch**, Stamford, CT (US)

**Reexamination Request:**
No. 90/009,978, Dec. 27, 2011

**Reexamination Certificate for:**

| | |
|---|---|
| Patent No.: | **6,038,342** |
| Issued: | **Mar. 14, 2000** |
| Appl. No.: | **08/114,608** |
| Filed: | **Aug. 31, 1993** |

**Related U.S. Application Data**

(63) Continuation of application No. 07/914,120, filed on Jul. 15, 1992, now abandoned, which is a division of application No. 07/799,549, filed on Nov. 27, 1991, now Pat. No. 5,278,918, which is a continuation of application No. 07/230,847, filed on Aug. 10, 1988, now Pat. No. 5,131,053.

(51) **Int. Cl.**

| | |
|---|---|
| ***G06K 9/20*** | (2006.01) |
| ***G06K 9/62*** | (2006.01) |

(52) **U.S. Cl.** ....................................................... **382/173**

(58) **Field of Classification Search** ........................ None
See application file for complete search history.

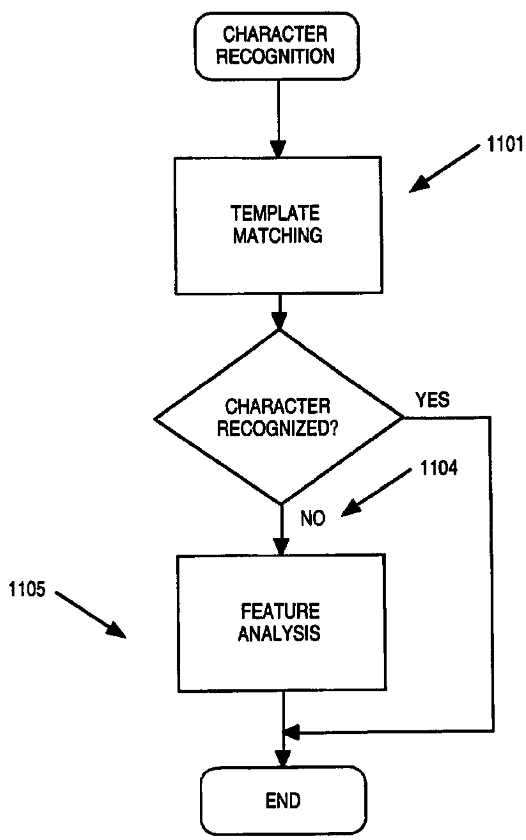
(56) **References Cited**

To view the complete listing of prior art documents cited during the proceeding for Reexamination Control Number 90/009,978, please refer to the USPTO's public Patent Application Information Retrieval (PAIR) system under the Display References tab.

*Primary Examiner*—Jason Proctor

(57) **ABSTRACT**

A system for recognition of characters on a medium. The system includes a scanner for scanning a medium such as a page of printed text and graphics and producing a bit-mapped representation of the page. The bit-mapped representation of the page is then stored in a memory means such as the memory of a computer system. A processor processes the bit-mapped image to produce an output comprising coded character representations of the text on the page. The present invention discloses parsing a page to allow for production of the output characters in a logical sequence, a combination of feature detection methods and template matching methods for recognition of characters and a number of methods for feature detection such as use of statistical data and polygon fitting.



# EX PARTE REEXAMINATION CERTIFICATE ISSUED UNDER 35 U.S.C. 307

NO AMENDMENTS HAVE BEEN MADE TO THE PATENT



AS A RESULT OF REEXAMINATION, IT HAS BEEN DETERMINED THAT:

The patentability of claims **1**, **4**, **6**, **7**, **13**, **14** and **18** is confirmed.

Claims **2**, **3**, **5**, **8-12** and **15-17** were not reexamined.

* * * * *

## CERTIFICATE OF SERVICE

I hereby certify that I electronically filed the foregoing with the Clerk of the Court for the United States Court of Appeals for the Federal Circuit by using the appellate CM/ECF system on November 14, 2014.

I certify that all participants in the case are registered CM/ECF users and that service will be accomplished by the appellate CM/ECF system.

Dated: November 14, 2014 /s/ Deanne E. Maynard

## CERTIFICATE OF COMPLIANCE WITH RULE 32(a)

This brief complies with the type-volume limitation of Rule 32(a) of the Federal Rules of Appellate Procedure because it contains 10,326 words.

Dated: November 14, 2014

/s/ Deanne E. Maynard